MECHANICAL DESIGN AND OPTIMIZATION OF A STANDARDIZED CUBESAT

A Thesis

by

DAVID PAUL MALAWEY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Reza Langari
Co-Chair of Committee,	Joseph Morgan
Committee Member,	Douglas Allaire
Head of Department,	Andreas Polycarpou

May 2016

Major Subject: Mechanical Engineering

ABSTRACT

Cubesats are tiny space exploration vessels with a lower cost than most satellites in history. Rapidly, studies are taking place on this technology to the degree that published papers on cubesats seem to have doubled between the initial background study and thesis completion. These satellites have a multitude of purposes and missions but all conform to one small physical form factor. The research presented here aims to create a cubesat mechanical design that is flexible enough to accommodate varying payloads and is fed by a multidisciplinary design optimizer in order to make best use of the limited volume of the craft.

The study consists of the development of a simplified math model of the satellite and subsystems, 3D model with FEA analyses based on successful past designs, and the creation of a real prototype using a CNC mill along with measures taken to reduce time and cost of prototyping. The results of the optimization suggest that a cubesat is a valid candidate for an optimizer which uses both heuristic and gradient based algorithms.

The optimizer first finds the satellite design with the lowest mass to meet the mission criteria while satisfying constraints on propulsion, electrical power, and structure strength. The design is a product of six variables (propellant, thruster valve, structure material, solar panel quantity, battery quantity, and structure thickness). Following the problem of mass minimization, cost is introduced in a dual objective optimization and a Pareto front of design options is generated.

Prototype development resulted in major improvements to prototyping speed and cost for the conditions at hand. The methodology and assumptions shown enhance the argument that these conditions are conservative with respect to most cubesat design requirements including strength, and manufacturing cost, and other metrics. Both the optimizer and the modular prototype of the spacecraft maintains an ability to manipulate the design in future iterations to offer an effective standardized platform.

Throughout this thesis, key lessons are shared and weaknesses and uncertainties are identified. Finally, future work is offered on this research topic which would enhance the effectiveness and certainty for success in full cubesat design implementation.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

A cubesat is a standardized type of small satellite with a form factor developed at Cal Poly in 1999 [1]. To date, hundreds of cubesats have flown in Low Earth Orbits, usually with missions that terminate after a couple of months as they combust in the atmosphere. Researchers and students have designed their own experiments to fly inside of cubesats but all cubesats have some common requirements including a sound frame, power sources, communications modules, and possibly attitude control. Figure 1 shows a typical cubesat. Some cubesat suppliers have sprouted in the industry but there is still room for design improvements and cost reduction.

Figure 1. Typical 1U Cubesat

2. PROBLEM STATEMENT

The purpose of the design project is to create a new CubeSat standard mechanical platform which meets targets to best serve a typical customer in the pico-satellite industry. Among these targets are payload capacity, cost, manufacturability, structural robustness, ability to accommodate typical cubesat components, and the requirement to fit within a standard envelope (Figure 2). The targets will be driven by benchmarks from existing cubesat designs, and existing empirical data will be used to support estimates of feasibility throughout the process. Design optimization methods are needed to minimize mass and cost while maximizing selected targets.

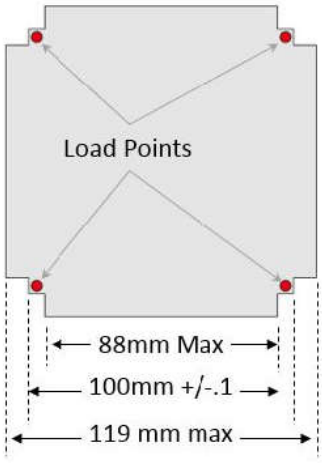


Figure 2. Design Space Dimensions

The cubesat design must accommodate future changes in design parameters because each cubesat mission has different needs. For example, some missions require a 2-U layout (as shown in Figure 3) or some may need additional power for high-wattage electronics. The cubesat industry is also being continuously fed with improved components which may change design constraints.

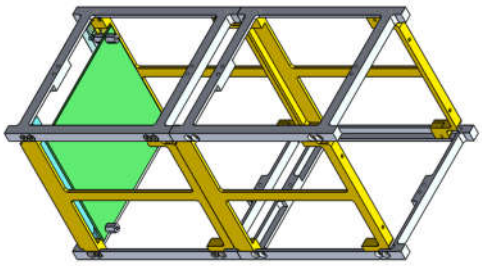


Figure 3. 2U Configuration

Design for flexibility can be accomplished in two ways. First, the physical framework should be modular and flexible in accommodating different configurations. Second, an optimization tool should be developed in a way that can be rerun with new parameters or adjusted to run under new constraints. The several interacting design variables such as frame material, propellant mass, flight attitude with respect to the sun, and cube outer geometry can be chosen concurrently using Multidisciplinary Design Optimization (MDO) methods.

Finally, a key aim of this design is to take advantage of additional volume available in the deployment apparatus in excess of the standard CubeSat size. While the CubeSat 1U universal standard is 100x100x100mm in external dimensions, there is additional space allowed by the NanoRacks deployment apparatus that can be used to benefit the user (as shown in Figure 2), given by the NanoRacks Interface Control document [8]. This design process aims to find the best way to use the extra space. Table 1 is a list of subsystems that need to be accounted for in the CubeSat design. This is not proposing to design all components but to be able to accommodate them in the mechanical design:

Table 1. Elements/Modules to Accommodate in Design

Element	Consideration
Antennae	Geometry, mounting location, inertia
Payload	Mounting strategies, power requirements
Power storage	Mounting location, volume (dependent on power requirements)
Solar panels	Deployment method, volume
Deployment mechanisms	Robustness, standard off-the-shelf components, lightweight
Attitude Determination Components	Mounting points for sun sensors, gyroscope, magnetometer
Attitude control (passive)	Permanent Magnets, weight boom
Attitude control (active)	Propulsion method, system mass, moments of inertia

3. LITERATURE REVIEW

3.1 Mechanical Configurations of Cubesats

In many design reports such as Dolengewicz et al, [4] several concepts were compared and selected from using the same goals as ours (cost, manufacturability, internal volume, structural soundness, modularity). Rather than repeating these full evaluations, we want to use the results to take a similar direction as those found to be the best designs. Many of these designs were also tested in FEA software and put through swept vibration tests. Resilience against vibrations is a major requirement of the CubeSat given by NanoRacks. However, rather than focusing on extensive simulations, the aim is to make structures that are mechanically similar to those already successfully tested.

The design of the frame deserves a close look before beginning the design process. Photos of other designs are limited and show very little detail on the frame. So, CAD models from other cubesats were downloaded when they were made available by the makers.

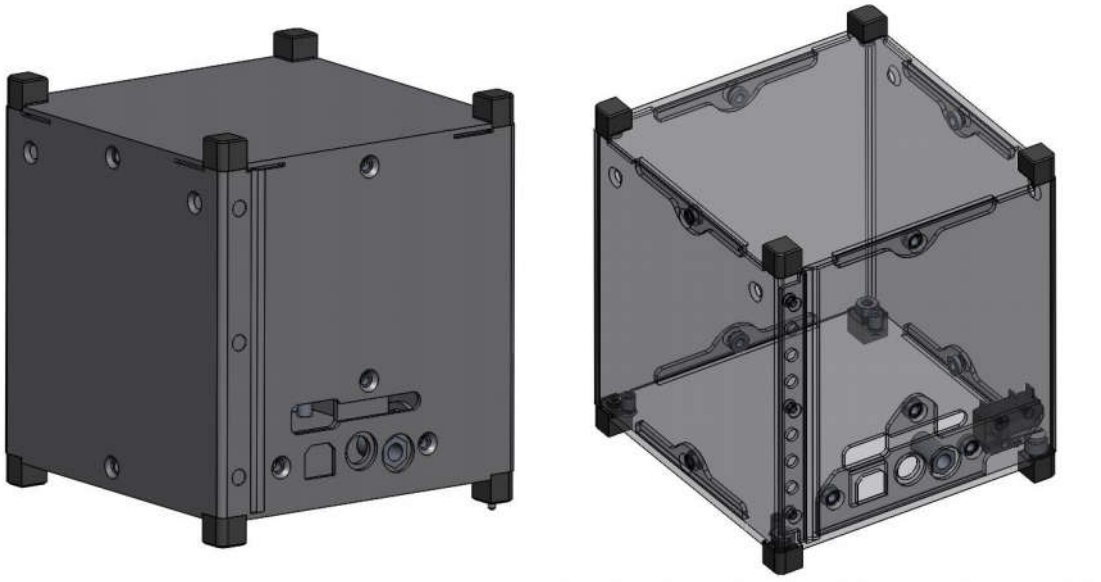


Figure 4. CubeSatKit Frame Design

The CubeSat Kit design, as shown in Figure 4, appears to be made from stamped metal, with thin walls that act as a frame. Sheet metal is not desirable for my design because the tools to create prototypes using sheet metal are not available. Making bends by hand using a metal brake in the machine shop would not meet tolerances near 0.1mm. Additionally, my design aims to expand the panels on each side, which would create complicated stresses and possibly requiring a deep draw die process to achieve.

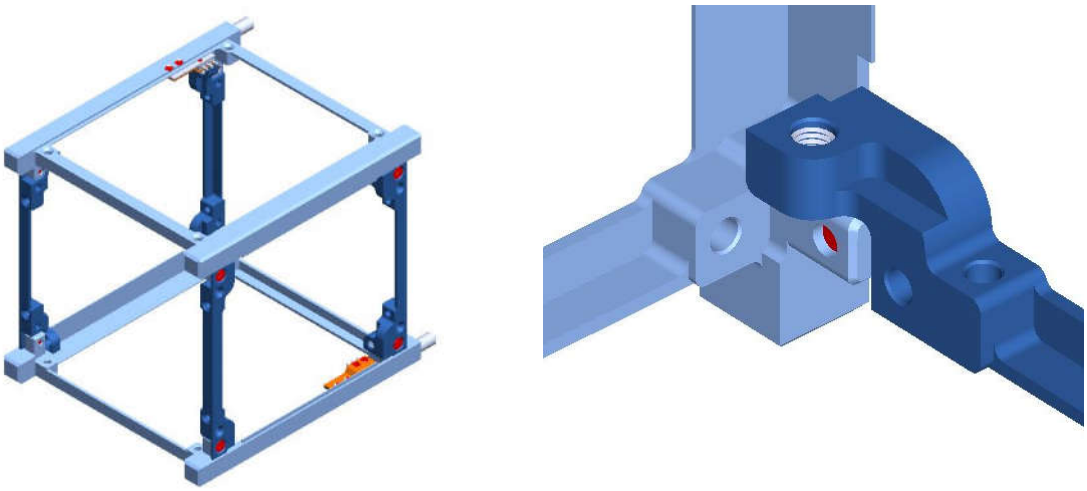


Figure 5. ISIS Frame Design

The ISIS Frame Design (Figure 5) is an attractive starting point for benchmarking. The frame is made from only two unique parts, which can be machined using a CNC mill. The mass appears to be minimal and the components of the frame can easily be evaluated for stress using FEA software, because of their similarity to a simple beam. However, the feature shown on the right side of Figure 5 cannot be made using a three axis mill, unless the part is rotated and refixed in the machine.

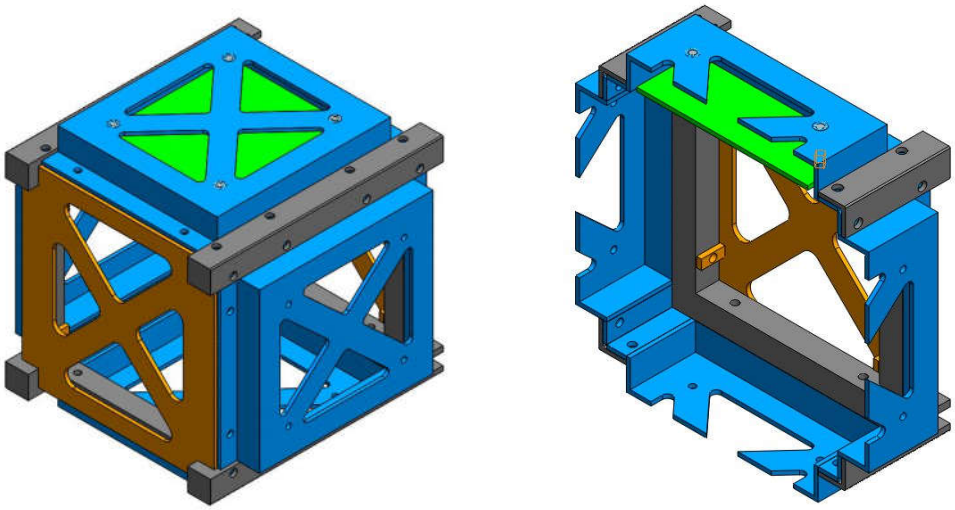


Figure 6. Cubesat Shared by NanoRacks

The cubesat design shown in Figure 6 was shared with me by NanoRacks but the designer is unknown. The design is enhanced in that the exterior panels fill the whole volume of the P-Pod envelope. However, the 3D model alone does not account for machining of any type. The cross section shows sharp exterior corners

that would be impossible to achieve using stamping methods and sharp interior corners that would not be achievable by machining methods. This design has an envelope I would like to imitate but offers no insight to a manufacturable design.

3.2 Component Designs

3.2.1 Propulsion

Most existing CubeSat designs, if equipped with attitude determination and control systems (ADCS), use magnetorquers and reaction wheels as actuators. The magnetorquers interact with earth's magnetic field to develop small torques and the reaction wheels use a spinning mass to change the angular momentum of the satellite. A minority of cubesats have taken the cold-gas thruster approach. These thrusters expel a compressed gas such as nitrogen or CO2 through a nozzle for momentum. Saint Louis University's "Rascal" 3U cubesat claims to achieve 6 degree-of-freedom movement using R134A refrigerant as propellant [12].

Figure 7. NASA Mini AERCam

Although most spacecraft using cold gas thrusters are much larger, it is a valid approach to attitude control in a cubesat. Arguably cold gas systems have only been kept from cubesats because of a pressure constraint in Cal Poly's Cubesat Design Specification (CDS) [2] which many cubesats follow. A notable small spacecraft using cold gas is NASA's Mini AERCam described in [3] and shown in Figure 7. On a separate small satellite, Assad Anis details another propulsion design using cold gas in his thesis [4].

Another possible attitude control method is passive stabilization. Rawashdeh explores a full passive attitude stabilization design in his master's thesis [5]. Some combination of passive stabilization and active control may offer more longevity for the mission.

3.2.2 Communication

Most cubesats have deployable antennae for transmitting data to a ground station. The mechanical design will not include the communication circuit but it does include the deployable antenna. A master's thesis by Sandvik [11] focuses heavily on the design of a tape spring antenna which deploys from a bent state into a straight line just like a tape measure. This kind of design is necessary to keep the antennae within the specified dimensions until post-launch.

3.2.3 Solar Arrays

Solar Panels are the primary means to produce more energy for a satellite. Using deployable arrays can increase the surface area available to absorb this energy, and most designs use some form of deployable array, as shown in Figure 8. The proposed design does not include a deployment design but must allow for some type of deployable solar panel. Scores of deployable panel designs can be found in existing cubesat documentation.

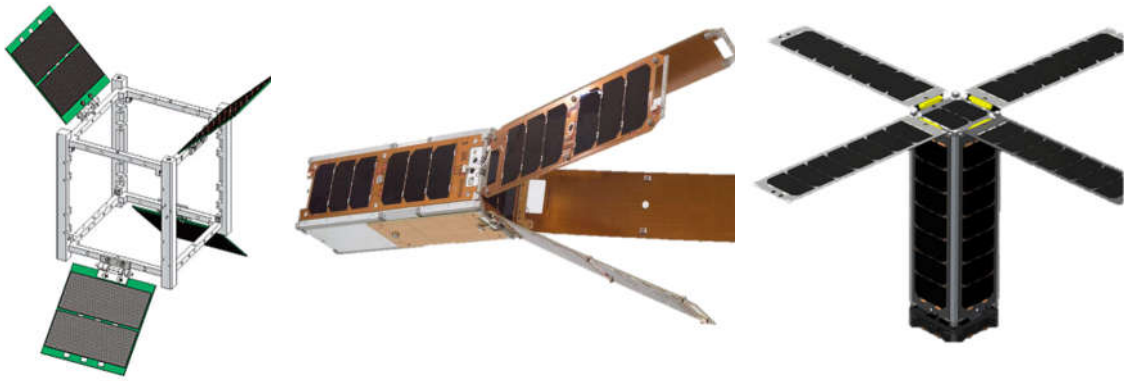


Figure 8. Solar Panel Expansion Configurations

The deployable solar panels featured in Figure 8 were adapted from Cajun Advanced Picosatellite, Pumpkin, and Gomspace sites respectively. The driving requirement for the deployables is that they must remain positively locked inside the PPOD envelope until triggered by the C&DH module post-deployment.

3.2.4 Attitude Determination

Kjellberg [6] lists and summarizes several instruments to include in an attitude determination system for cubesats at the University of Texas. He offers a trade study with characteristics of sun sensors, magnetometers, and gyroscopes, and makes a recommendation of the most suitable components. Similar components will need to fit into the proposed cubesat structure.

3.3 Optimization in the Cubesat Field

CubeSat Design reports using an MDO approach are scarce. Among these, Hwang et al, at the University of Michigan published a detailed design project in an article titled "Large-Scale MDO of a Small Satellite

using a Novel Framework for the Solution of Coupled Systems and their Derivatives" [7]. This article, along with its optimization tool, offers a thorough MDO setup for optimizing CubeSat designs. The tool is free to the public but it does not include a module for cold gas propulsion. So, implementing a cold gas propulsion module within a cubesat MDO is an area open for advancement.

The other major publication on MDO for cubesats is titled "Design Optimization of a Solar Panel Angle and its Application to Cubesat 'CADRE" [8]. This paper describes an optimizer designed to utilize the interaction of solar incidence and flight characteristics, which are both effected by solar panel angle, to increase power generation and ultimately maximize transmitted data. The optimizer focuses on an existing cubesat design and manipulates the configuration of this satellite rather than using the optimizer to specify all of the components.

Given the existing cubesat optimizations, there is merit in creating an MDO code which incorporates nearly all subsystems in the model. This design will cover a wide range of elements and have many possible failure modes, it is beneficial to refer to previous missions and their outcomes. A journal article by Michael Swartwout statistically summarizes the first 100 cubesat missions [9]. Considering this article should help to avoid common shortcomings to mission success.

4. METHODOLOGY

4.1 Research Process

1) Benchmarking of existing CubeSat platforms for structure, mechanical functions, and main design parameters.
    a. Break down the designs into "modules" that can be directly compared and quantified.
    b. Download 3D CAD files for available models and compare geometry and method of securing the payload.
    c. Develop metrics and targets by which to optimize a design
2) Develop an MDO program to optimize mechanical design & functionality and exploiting interactions of different components.
    a. Use/make simple math models with justified assumptions to model the interaction of design variables (i.e., solar flux vs cube orientation).
    b. Continuously share design information between the mechanical design and electronics design addressed by the capstone team using documentation.
    c. Choose a mechanical configuration which will best meet the design targets.
    d. In this process, all requirements including manufacturability must be held.
3) Understand the satellite environment including shock, vibrations, temperatures and electromagnetic interference and account for these elements if possible.
    a. Identify failure modes and first search for equivalent existing designs that have been validated by testing.
    b. Where appropriate, run simulations to verify module designs.
    c. As a last resort, perform physical testing where there is not enough certainty in the simulations or comparable designs.
    d. Report on factors that cannot be confirmed or require future testing.

4.2 MDO Execution

The optimization problem is a minimization of a function, J, of multiple design variables with bounds on the variables and relevant constraints applied to it. Papalambros et. al. [10] describes the general problem setup.

$$\text{minimize} \quad J(x,p)$$

$$\text{subject to} \quad g(x,p) \leq 0$$

$$(x, p) = 0$$

$$x_{i,LB} \leq x_i \leq x_{i,UB}$$

$$\text{where} \quad J = [J_1(x) \ldots J_z(x)]^T$$
$$x = [x_1 \ldots x_i \ldots x_n]^T$$

In this formulation, X is the design vector and P is a set of design parameters. Constraints are divided into g (inequalities) and h (equalities). And the bounds are set by LB and UB for each design variable x. Starting from this point, some x and p values may be interchanged. For instance, if a variable is chosen to be fixed in the design, it can become a parameter.

Upon execution, a combination of gradient based methods and heuristic methods may be required. Some continuous functions may be given for variables like "propellant mass" but discrete vectors such as "number of battery cells" will make the design space unsmooth. Lastly, the problem can be expanded into a multi-objective function such that multiple targets can be minimized such as mass, cost, or inverse of power lifetime.

4.3 Outputs

The following outputs were stated upon the proposal of the research topic.

1. Benchmarking data, quantified design targets and explanation of how the design is competitive.
2. Software-based optimization code which integrates a cold gas propulsion system.
3. 3D CAD models of mechanical parts including:
    a. Frame components and fasteners
    b. Deployable antenna
    c. Mounting points for main circuit board(s)
    d. Deployable solar panel attachment
    e. Mounting points for main components: batteries, propulsion units, etc. (ultimately considered to mount with c because these accessories have not been developed)
4. Results of simulations and testing that accounts for identified failure modes.
5. A fabricated prototype of the CubeSat frame.
6. Instructions (drawings) for manufacturing the product in a way that will meet the original design intent. These are found in Appendix B – Mechanical Drawings and Appendix C – Fabrication Materials

5. MULTIDISCIPLINARY OPTIMIZATION

The following section describes the process of developing and evaluating the cubesat design as a multidisciplinary design optimization problem.

5.1 Model Formulation

5.1.1 Nomenclature

X	=	design vector	x^*	=	optimal design vector
X_0	=	initial design vector	λ	=	weighting factor
J	=	objective value	I	=	moment of inertia
g	=	nonlinear constraint			

5.1.2 Design Vector

The design vector is described in Table 2. The first three variables are strictly discrete and have a table of properties associated with each selection. Solar panel quantity and battery quantity are discrete but can be treated as continuous in most of the project. Structure rail width refers to the cubesat frame and is continuous.

Table 2. Design Variables

Variable	Description	Metric	Lower	Upper
X(1)	Propellant type	Gas type	1	9
X(2)	Thruster type	Model	1	3
X(3)	Structure material	Material	1	3
X(4)	Solar panels	Quantity	0	4
X(5)	Batteries	Quantity	1	Inf.
X(6)	Structure rail width	(mm)	3	20

The bounds (given by the last two columns) are driven by available data for the first three variables. More materials, propellants, and thruster models can be populated into the problem if data becomes available. The number of solar panels is limited to a maximum of four by space available on the sides of the satellite, although it is not impossible to have more. There must be at least one battery for power storage when sunlight is not available to the solar panels. Lastly, the structure rail width has a minimum of 3mm so it is practical to create threaded holes in the rail for fasteners.

5.1.3 Constraints

The satellite must have enough electric power, structural rigidity, and propellant to complete the mission described by the system parameters. The constraints are listed in Table 3. The constraints listed as hard-coded mean that the constraint is resolved inside a module. For example, the propellant module takes in the

properties of the propellant specified by the design vector and generates a value for how much propellant mass is needed. Since this is a deterministic solution we choose not create a "propellant mass" design variable and waste computation time on designs which may not meet the constraint. Power generated will be a combination of batteries and solar panels, while bending rigidity is a function of material and structure geometry.

Table 3. System Constraints

Effect of Constraint	**Type**	**Bound**
Power generated is sufficient	Inequality	> Power consumed
Bending rigidity is sufficient	Inequality	> baseline rigidity
Propellant amount is sufficient	(hard coded)	= mass required to counteract flight disturbances
Prop tanks qty is sufficient	(hard coded)	= enough to hold propellant volume

The constraint for the battery/power module is shown in eq 5.1. In summary, the power drawn by all electronic components for their given duty cycle over the duration of the mission must be met or exceeded by the power stored in the batteries and generated by the solar panels.

$$\sum_1^n (I_n v_n D_n \quad 24P_1) \quad (X_5 v_b P_7 + X_6 P_3 P_8 \quad 24P_1) = 0 \tag{5.1}$$

I_n= current
v_n= voltage
D_n= duty cycle
P_1= mission duration
n = no. devices
v_b= batt voltage
P_7= batt capacity (Ah)
P_3= panel power rating
P_8= sunlight ratio

5.1.4 N^2 Diagram

Figure 10 is a simplified representation of our N^2 diagram. Feedback has been reduced to only one pair of modules where a fixed-point iteration method is used to make the prop module and battery module agree on propellant and power amounts. The mass and cost data is actually computed in one module but for single-objective optimization, only mass is used.

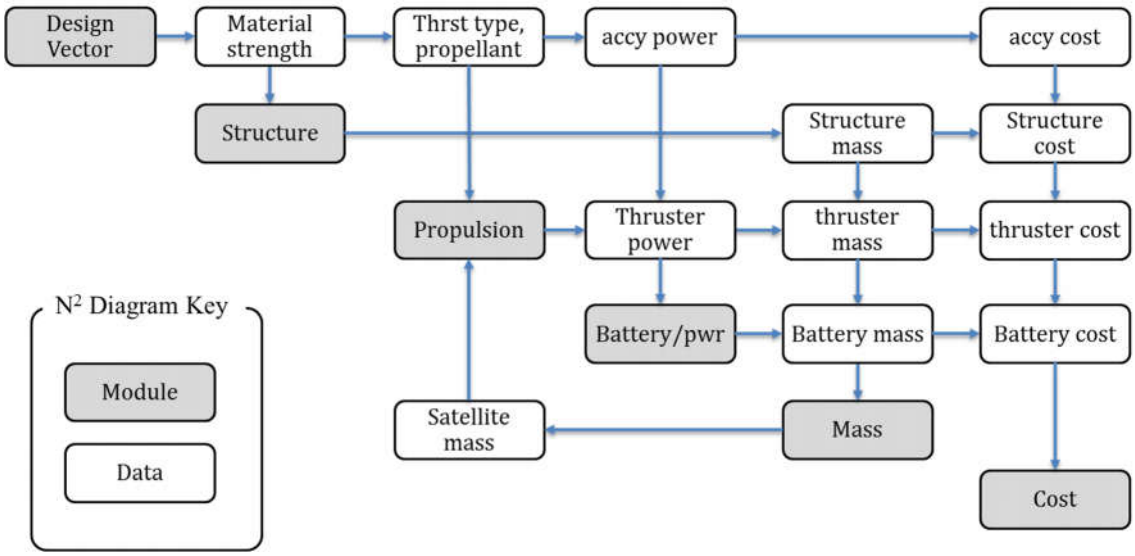


Figure 9. Full N^2 Diagram

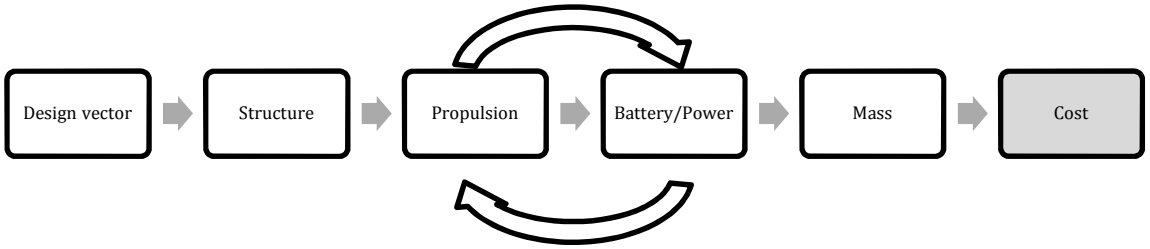


Figure 10. Module Sequence

5.1.5 Structure Module

The structure module revolves around the X(3) and X(5) design vectors, structure material and rail width. The most important part of the structure that can be modeled simply is the rail, which gives rigidity to the satellite. Figure 11 (a) indicates four of the eight rails of interest.

The constraint associated with this module is that the stiffness of the final design must be equal to the baseline stiffness (given by modulus *moment of inertia of the rail cross section). The cross-section is represented in Figure 11 (b). This baseline value comes from existing satellites that have flown successfully.

The principal moments of inertia are along the “a” and “b” directions, with “a” being the maximum. This area moment of inertia is described in equation 5.2.

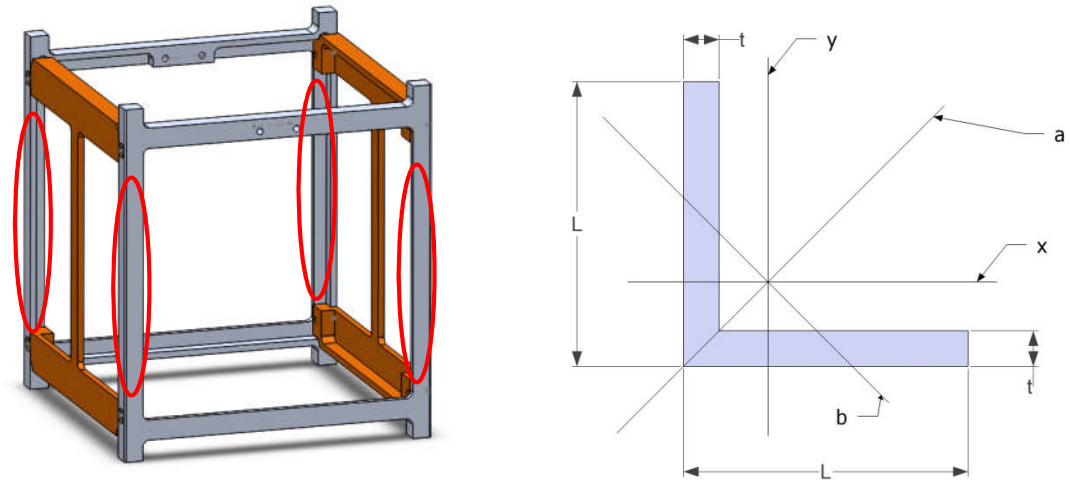


Figure 11. Cubesat Frame (a) CAD Model (b) Rail Cross Section

$$I_a = \frac{t(2L-t)(2L^2-2Lt+t^2)}{12} \tag{5.2}$$

In the structure module, the rail’s cross section is assumed to be a square L beam with thickness t fixed as a parameter at 2mm. The purpose of this minimum is to protect against buckling and allow for fasteners to be secured through holes in the material. If a flexible material is selected in X(3), the rail width “a” will need to increase to meet the rigidity constraint. Finally, the cross sectional area is calculated and combined with the material density to give a new structure mass. The mass is then used by the following modules.

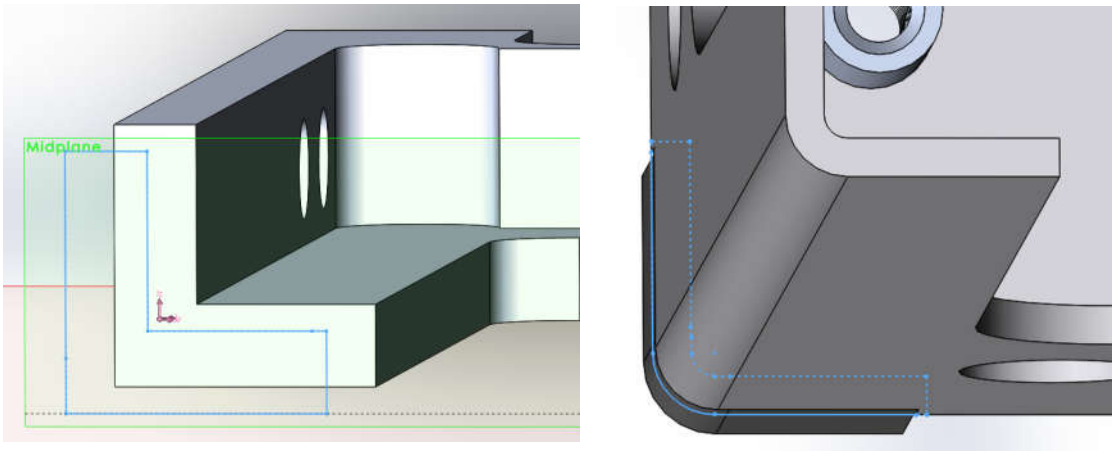
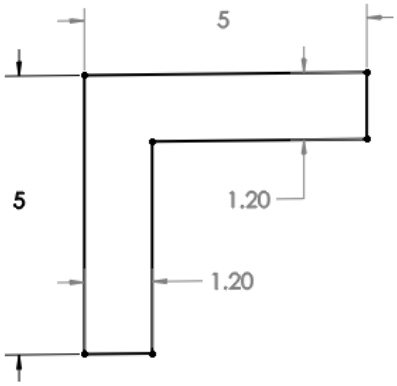
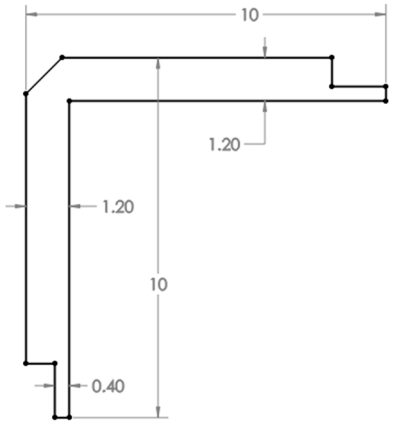


Figure 12. Structure Rail Sections for Sidepanel (L) and CubesatKit (R)

The purpose of the structure module in the optimizer is not to fully model the strength of the frame but to demonstrate the ability to compare the strength of the design to a benchmark. Figure 12 shows that the Sidepanel and the chassis from Cubesatkit.com share this same general feature.

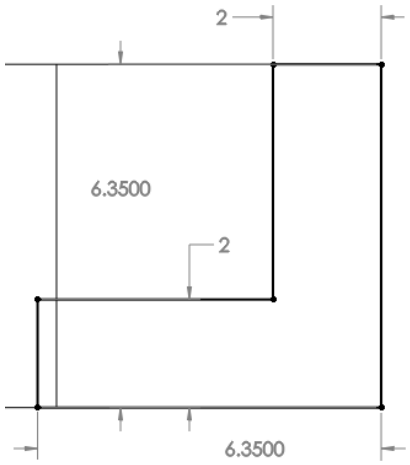
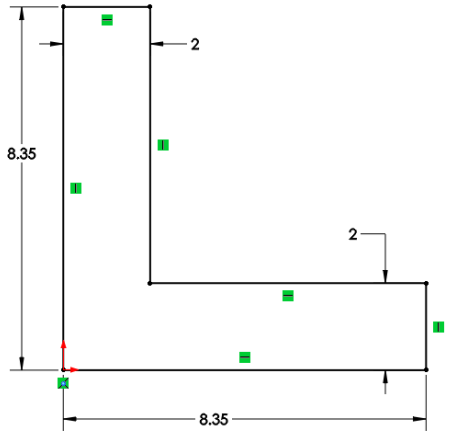
After inspecting multiple benchmark designs, the ISIS chassis was chosen for the benchmark for the optimization baseline. This design is most similar to my design and it includes two different styles of rails that make up the chassis. The I_a and I_b for these rails are shown in Table 4.

Table 4. Benchmark Cross Section Data

ISIS rail 1: (x4 per cube)	ISIS rail 2 (x4 per cube)
	
$I_a = 34.7\ mm^4$	$I_a = 239\ mm^4$
$I_b = 10.1\ mm^4$	$I_b = 61.2\ mm^4$

The Sidepanel initial design was created early in the process for kicking off the prototyping work. After a comparison of my initial design to the benchmark, it was decided that the safest method is to make a rail that sets the benchmark's *stronger* rail moments as the minimum for *all* designed rail moments. A revision to the initial design is then proposed as indicated under the "Sidepanel Modified Design" shown in Table 5. This has both minimum and maximum principal moments that exceed the benchmark's stronger rail.

Table 5. Designed Cross Section Data

Sidepanel Initial Design	Sidepanel Modified Design
	
$I_a = 106\ mm^4$	$I_a = 270\ mm^4$
$I_b = 34.3\ mm^4$	$I_b = 78.4\ mm^4$

The mass of the structure subsystem is also an impact of the structure design variables and is calculated by the following methodology. Using Solidworks, the initial mass of a design is measured. Solidworks uses the volume of the frame and the material density to make this calculation. The structure module makes the simplifying assumption that the total material of the frame is proportional to the area of the frame cross section. So, the final mass is calculated by using eqn 5.3, where A represents the area of the cross section.

$$\mathbf{m_{final}} = \mathbf{m_{initial}} \ \frac{\mathbf{A_{final}}}{\mathbf{A_{initial}}} \tag{5.3}$$

After equation 5.3 is applied, the mass is adjusted again by the density of the chosen material if a new material has been chosen.

5.1.6 Battery/Power Module

The battery module uses design variables X(4) and X(5), quantities of batteries and solar panels, to calculate the electric energy available to the cubesat and compare it with the requirement. Batteries of course have a fixed capacity at full charge and solar panels will offer more energy for each day of the mission. For mass the solar panels a strong advantage as a method of supplying energy but they are far more expensive.

In the module, the power consumption and duty cycles of multiple components are added to find the required energy. Parameters such as percentage of orbit spent in sunlight are estimated and can be adjusted to improve real-world accuracy. The two design variables are discrete, with zero as a lower bound for panels and 1 as a lower bound for batteries. Lastly, the electric system mass is added up and sent into the propulsion module.

Specifications for the batteries in the optimizer comes from a Lithium Ion 18650 battery made by Panasonic (Figure 13, left). This style of battery appears to be an industry standard when researching power supply modules designed Specifically for Cubesats. The Gomspace power module shown in Figure 13 (right) uses this style of battery as do other popular satellite power options. The selected battery has a capacity of 3400mAh and a nominal voltage of 3.6v. The batteries were ordered and weighed at 46 grams. This data is used in the optimizer for power and mass calculations.

Specifications for the solar panels are adopted from Clyde Space solar panel specifications found in [11]. In summary, the solar panels weigh 45 grams and produce a power of 2.1 watts per 1U square, at a nominal temperature of 28 degrees C. Any reduction in solar angle of incidence or reduction in temperature causes a decrease in the power generated. According to [12], the standard testing for this type of solar cell involves beginning-of-life values "…measured at 28°C, under the AM0 spectrum (solar irradiance spectrum in space), and assuming solar flux of 135mW/cm2."

Figure 13. Optimizer Battery

5.1.7 Propulsion Module

For a successful mission the satellite needs to be oriented properly in space. Primarily the attitude must be correct to aim the antennas and ensure communication back to earth. The propulsion system must offer enough kinetic energy to keep the satellite oriented for the duration of the mission with some disturbances introduced each day. For cubesats, "delta-v" is the metric to describe how much kinetic energy can be imparted by its Attitude Control System. For this design the cubesat will not change velocity, only orientation.

Our required delta-V is a guess based on benchmarking similar systems and it increases if the length of the mission is increased. If the mass of the satellite increases then more propellant is required to achieve the same delta-V. This mass is calculated in the module.

Table 6. Propellant Properties

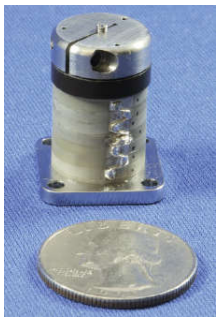
Propellant	Density (3500 psia, 0C) (g/cm3)	Specific Impulse (s)	Cost($/kg)
Hydrogen	0.02	296	120
Helium	0.04	179	52
Neon	0.19	82	330
Nitrogen	0.28	80	4
Argon	0.44	57	5
Krypton	1.08	39	330
Xenon	2.74	31	1200
Freon 14	0.96	55	10
Methane	0.19	114	10
Ammonia	0.88	105	10

The propulsion module uses design variables X(1) and X(2), propellant type and thruster type. For propellant type, the interactions involved are the density of the propellant and the specific impulse of the propellant. The propellants used are listed in Table 6 with data that comes from [13]. With the propellant tank volume fixed as a parameter there is a tradeoff between a dense propellant which weighs more and a

light propellant which may require multiple tanks (adding more mass for each tank). For thruster type, the general tradeoff is that a large heavy thruster consumes less battery power in operation while a compact light thruster may require higher power.

The Lee Co

[14]



Busek PFCV

Moog

Figure 14. Thruster Valves from Four Manufacturers

The thruster characteristics, illustrated in Table 7, give insight to the tradeoffs one encounters in thruster selection. The data in this table is a combination of values taken from vendors' publications and estimations developed to make the model functional. Essentially higher performance leads to a higher cost, and in a fixed-cost scenario, the two performance metrics form a trade-off relationship. A highly efficient, low power-consumption electromagnetic coil is likely to weigh more while a highly compact, lightweight coil may require higher wattage to actuate.

In a final design, a closer study should be created instead of comparing these thrusters as equivalent replacements for one another. Some have the thruster outlet integrated in the unit, and some require this component to be added or other special mounting hardware. For the purpose of this optimization study, this data is sufficient to demonstrate an optimizer selecting the tradeoffs in varieties of this kind of component.

Table 7. Thruster Properties

Thruster	Mass (g)	Wattage (W)	Cost*
Lee Co	35	0.04	700
Marotta CGM	59	1.00	222
Busek PFCV	17	0.89	66
Moog	09	1.00	111

Since the propellant power is fed back to the battery module and the battery mass is used in the propellant module, a fixed point iteration is used to make these two modules converge.

Inside the propellant module, the cold gas is assumed to be stored in a commercial off the shelf style tank. The design comes from inexpensive and readily available compressed gas capsules such as those for powering airsoft guns. Since the tank is standardized, the number of tanks must be adjusted to accommodate the volume of gas required by the system. Figure 15 shows a visual for how the optimizer would select three tanks to store 35mL of cold gas. Each tank has an associated dry mass which impacts the objective function overall.

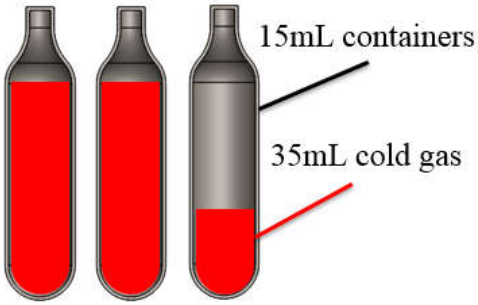


Figure 15. Propellant Tanks

The simplified propulsion model does not include calculations for all hardware in a propulsion system. A full propulsion unit will include extra pipes and hardware for assembly and fastening. It may include a heater to protect components from dropping below operating temperatures, and a thruster nozzle for each valve if the valve does not come with an integrated nozzle. It may also need a special power regulation components depending on the operating voltage and current drawn by the electromechanical valves. These elements would add mass and cost to the system.

Still, the model is valid in concept because these accessories would be required for all configurations chosen by the optimizer. So, when the optimizer chooses the design with the lowest mass, the final mass may be inaccurate but it should still be the minimum.

5.1.8 Cost (& Mass) Module

The cost module is last in line because it takes parameters from all modules and adds up the satellite price. Material, thrusters, propellant, batteries, and solar panels all have associated prices. Both raw material and machining costs are included. It's interesting to note that the cost of machining a hard material has a larger impact than buying an expensive raw material. Machining cost is calculated from a baseline estimate from a machine shop at $960 and the "machinability" factor of the material which was retrieved from [15].

Propellant cost is given by the mass of the propellant and the price per kg of propellant taken from [16]. Other component prices were found online from suppliers and manufacturers.

Some values for cost had to be fabricated, since the suppliers could not offer a price. This was due to valves requiring custom tailoring to the application, or a model that is no longer offered and a similar new model would need to be chosen with detailed discussions.

A simple model was made to fairly price thrusters with the assumption that each component had a tradeoff between cost, mass, and power consumption. This relationship is shown in eq 5.4 and expresses that the product of mass and power (both desirable to minimize) would have a constant relationship with cost.

$$\frac{Mass\ \ Pwr}{Cost} = Constant \tag{5.4}$$

5.2 Optimal Design by Mass Objective

The first optimal design was made to minimize mass. Having run several iterations, debugging the program, and updating parameters based on new data meant that one optimal design vector was not maintained throughout the work. The optimal design in this section refers to one of the latter optimal designs for the single-objective minimization result.

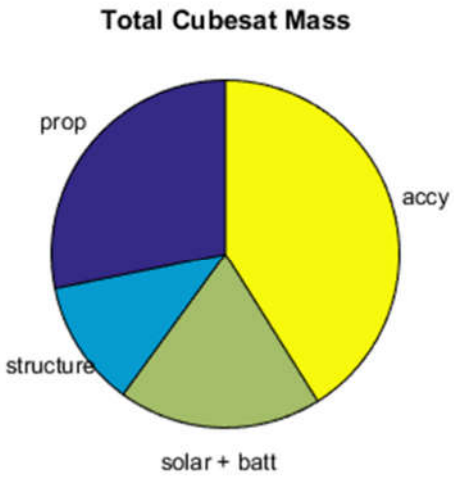


Table A: Design Vector

	Value
total mass (g)	803
propellant	Freon
thruster	Moog 1W
material	titanium
solar panels	2.3857
batteries	1
structure width(mm)	3.0402
cost($)	5134

Table B: Relevant Data

	Value
algorithm run time (s)	1
prop tanks (qty)	2
propellant mass (g)	30
pwr consumption (Wh)	373
solar pwr produced(Wh)	361

	prop	structure	solar + batt	accy
mass (g)	227	95	152	330

Figure 16. Optimal Design Graphical Output

The resulting optimal-mass design weighs 803 grams and is a repeatable result. For the design variables, Freon is the selected propellant, with the Moog brand thruster, titanium chassis and batteries minimized to one while solar panels are meet the remaining power demand.

Figure 16 shows a graphical output programmed into the Matlab code for the purpose of checking designs. It gives information about the design vector, as well as additional breakdowns of important metrics. For example, the power consumption constraint can be visually checked in that the total power consumption is 373 Watt-hours and nearly all of that is produced by the 2.4 solar panels on the satellite.

This graphical output is highly useful in troubleshooting the code and detecting if appropriate changes take place when parameters are changed. The number of propellant tanks are not an explicit design variable but it is important to see this increase from two units to accommodate more propellant when the mission duration is increased in the parameters.

This X* design and the associated parameters that deliver this result were taken as a baseline for further evaluation, including sensitivity analyses on both the design vector and the design parameters.

5.3 Sensitivity Analysis

5.3.1 Sensitivity to Design Vector

A sensitivity analysis gives more insight to the optimal design solution, and how the objective function is affected by small changes in the selected design. The following sensitivity analysis was performed about the X* design as follows: Xenon propellant, Lee thrusters, Titanium material, 1.5619 solar panels, 1 battery, and a 3.04mm structure width.

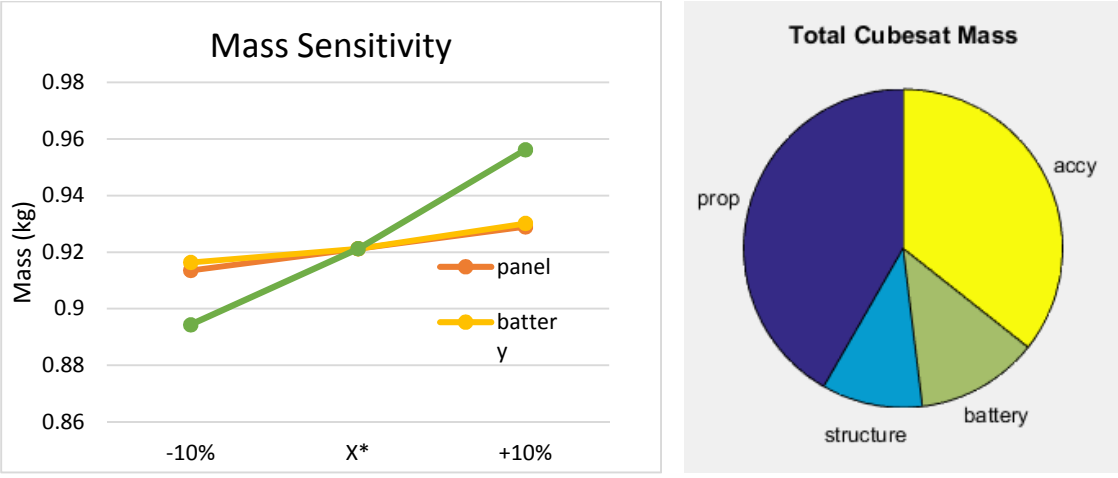


Figure 17. Cubesat Mass (a) Sensitivity Analysis (b) Subsystem Mass

Three variables were adjusted from the optimal design point and plugged back in to calculate the objective function. These three variables have most meaningful results for sensitivity because they can be treated as continuous. As a separate study, it may make sense to perturb a discontinuous variable such as propellant if there were a risk of that propellant becoming unavailable or disallowed for safety reasons. Figure 17 shows the impact of the three "continuous" variables on the resulting mass. This result is somewhat intuitive. First, all three items have a linear effect on the mass. Solar panels and batteries have a similar weight per item and have a similar quantity (>2) in the design. In Figure 17 (b), the structure is shown to carry about the same mass as the batteries and solar panels combined, which should lead to the slope being near double that of the other two items.

Using the same perturbations of X(4) through X(6), the effect on cost was observed. These results are shown in Figure 18 (a). The cost is insensitive to perturbations of the battery and structure variables. The batteries are the least expensive part of the satellite. The structure accounts for a significant portion of the cost (shown in Figure 18 (b), but it is mostly a fixed cost due to machining and minimally effected by the thickness of the rail. The solar panels on the other hand are expensive (priced at $1500 each in the current optimizer settings). So, the cost is most sensitive to solar panels of these three variables.

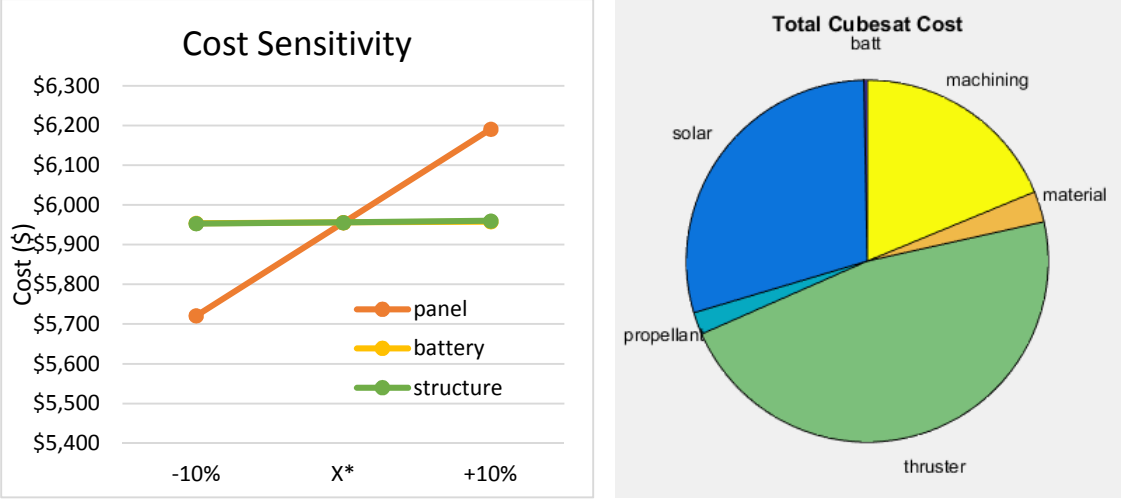


Figure 18. Cubesat Cost (a) Sensitivity Analysis (b) Subsystem Cost

Real world impacts of the sensitivity data should be considered. For example, if any of the three design variables were reduced by 10%, the design would no longer be feasible. The power would be insufficient for the mission or the strength of the frame would fall below the specified requirement. On the other hand, the information could be used to benefit us. Since we cannot truly implement a design with 1.56 solar panels, it may be possible to explore solar panels that are ½ size of the cube face (these are actually available)

and to make up the power requirement by adding one battery. This kind of adjustment is likely to be feasible while reducing cost and adding just a small amount to the mass.

5.3.2 Sensitivity to Parameters

The second part of the sensitivity analysis checks the effects of changing various parameters. In solving for the design vector the parameters are considered fixed, but they may change from time to time. To study the effect this may have, the dual objective function was evaluated for several new cases. The difference between this study and the last section is that the optimizer was actually re-run this time to take advantage of the large parameter changes and to ensure feasibility of the resulting design.

Table 8. Parameter Changes for Sensitivity Analysis

Parameter	nominal	new	Effect
accy mass (kg)	0.33	0.66	mass increase only 337g
thruster (qty)	4	2	change from Moog (9g,1.0W) to Busek (17g,0.89W)
struct L.bound (mm)	3	1	1.66mm A36 steel
batt capacity (mAh)	3400	5000	from 2.39 --> 2.35 panels
prop tank volume	16.3	30	Xenon 1 tank
disturbance/day (ΔV)	0.5	0.75	30g Freon + 2tank, 43g Freon +3tank
duration (days)	30	60	58g prop + 4 tank, pwr cons increase to 746 but made by solar
X*	-	-	freon, titanium, Moog1W

Seven parameters were adjusted in a way that could take place in the real world. For example, the mass of the payload (accessory) could be doubled, or the thruster quantity could be reduced to two if the ACS needed fewer degrees of freedom. All of these choices are summarized in Table 8. For each choice, the (highly summarized) effect on the design vector was recorded in the last column.

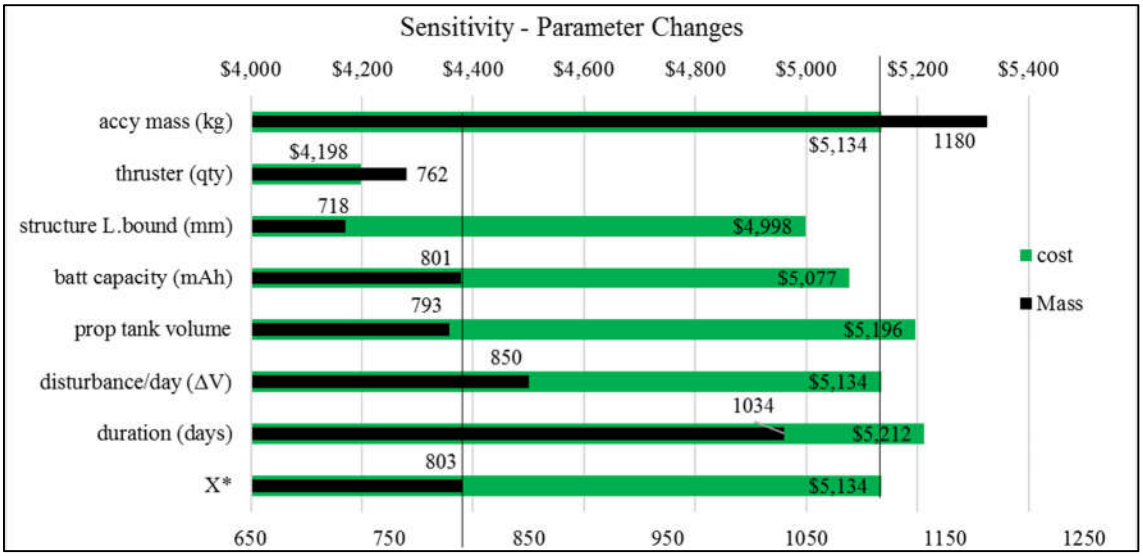


Figure 19. Sensitivity to Parameters

The impacts of the changing parameters are summarized in Figure 19. Two vertical lines represent the mass and cost of the nominal (X*) design by which the other results can be compared. In most cases, the parameter change gives a very evident impact in one of the two objective functions, and a minor impact on the other.

Increasing the propellant tank volume creates an interesting tradeoff. The new tank can hold *almost* enough Freon to achieve the mission with only one tank, so then the gas is changed to Xenon with a higher density and a higher price. The result is a reduced overall mass and an increased overall cost.

This part of the parameter study is an example of how the optimization could serve a customer in the real world. In the software, one propellant tank has a significant mass but costs less than $20. So the removal of this tank does not offset the high price of Xenon gas but does improve the satellite weight by 1.2%. In practice, the hardware and integration costs of having multiple tanks may be great enough that a cost savings is realized by this parameter change contrary to the parameter study results. If a larger COTS tank became available, it would probably be recommended to the customer based on this kind of study.

5.4 Pareto Front

The cost was added into the objective function such that the objective function became a weighted sum of cost and mass. This objective is illustrated in eq. 5.5 where lambda is a value between zero and 1 and raises on each iteration of the optimization by N equal increments. The Pareto front in Figure 20 is the result.

$$J = cost(\lambda) + mass(1 \quad \lambda) \tag{5.5}$$

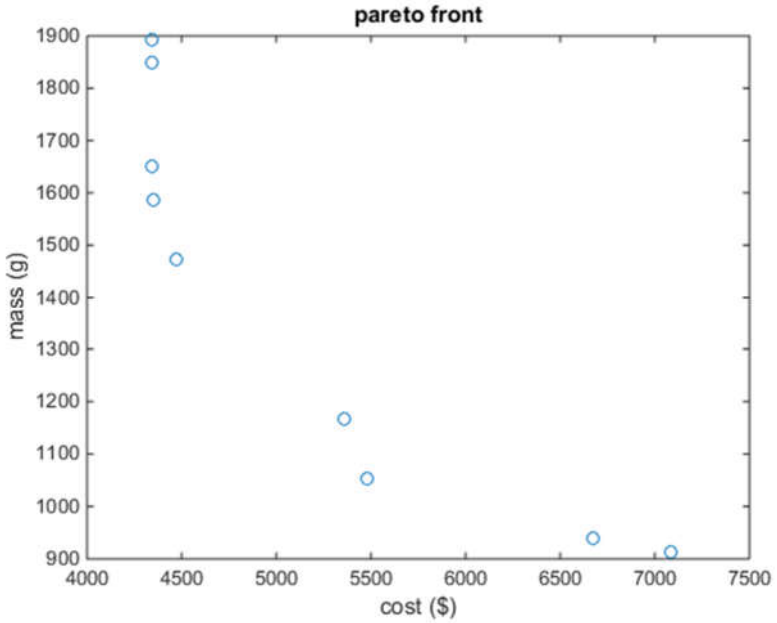


Figure 20. Pareto Front with 2-Stage Optimizer

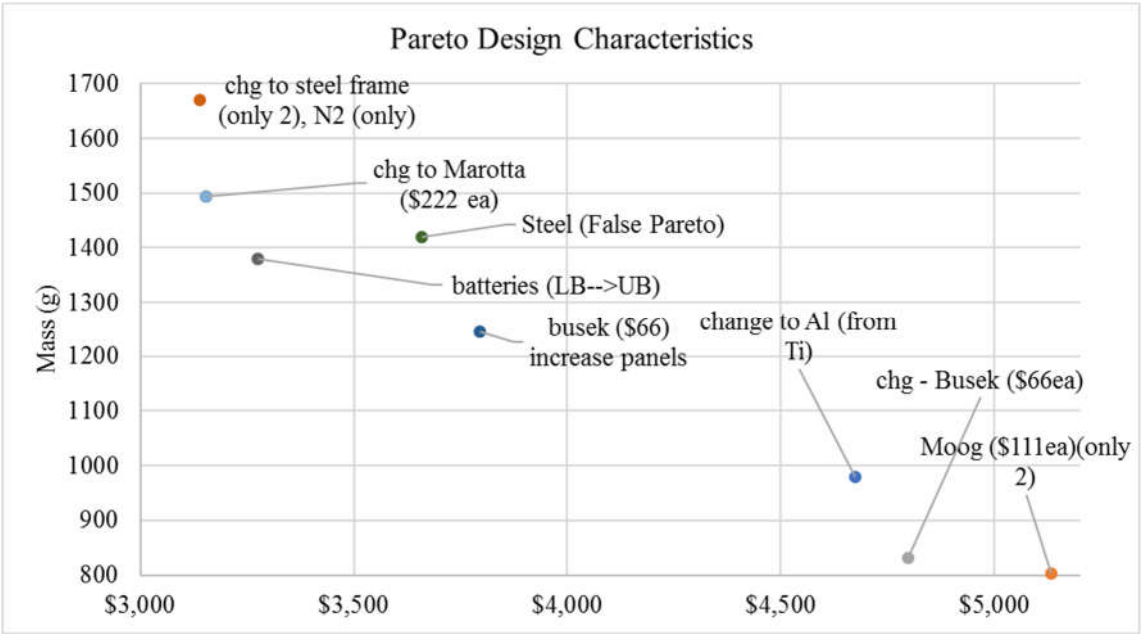


Figure 21. Pareto Front with Design Notes

A careful look at the Pareto optimal designs leads way to Figure 21 which characterizes each design by it's major change from the previous design. Considering the designs from most expensive to least expensive,

these shorthand notes describe the decisions made by the optimizer which change the cost and mass outcome. For example, just below the 1400-gram mark, all designs changed from forcing the battery quantity from the lower bound to the upper bound. Since batteries cost far less than solar panels ($10 vs $1500 off the shelf) but offer less energy per mass, this switch creates a major jump down in cost and up in mass.

In several of these designs, a major change in tradeoff such as battery vs panel is accompanied by reversing earlier tradeoff decisions which have a smaller impact. The result is a small step up the Pareto line instead of a major jump which would take place if all other variables remained fixed while making a major change.

Another notable result from the Pareto chart is an occasional false optimal design. Figure 21 features a design with a steel frame that is bested by other designs in both cost and mass. This takes place when the genetic algorithm fails to give a design result that is in the optimal region of interest and the gradient based method does not have enough power to leave this region because it can only control the last three variables.

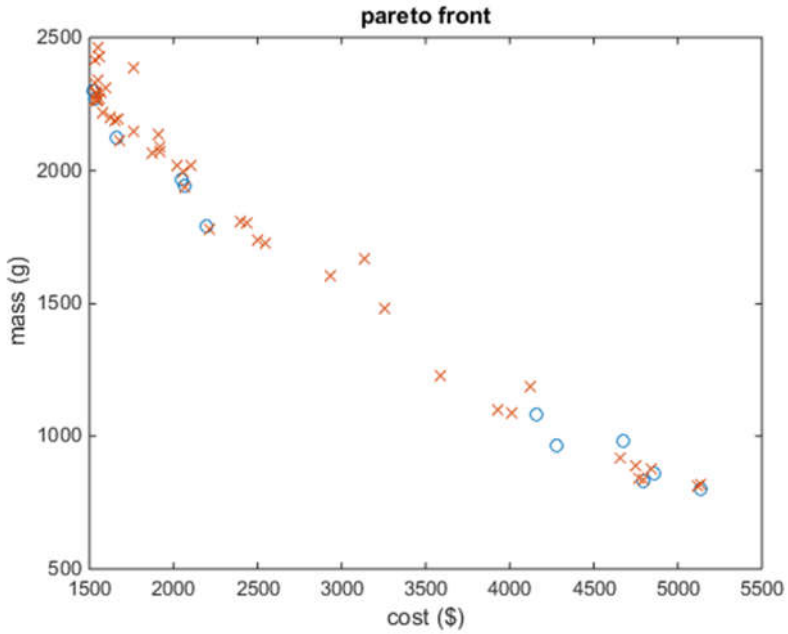


Figure 22. Pareto Front with 1 and 2 Stage Optimizer

One issue found in the first generated Pareto front was a somewhat bare population of results. Even when N was increased to 100 iterations, the final designs all converged to the same designs shown circled in Figure 20. When the whole program is rerun, the results only changes by a couple of design points. To investigate this problem, the study was rerun using only the genetic algorithm and plotted on top of the results from the original study.

This plot (Figure 22) shows that several unique designs are generated by the GA which are eliminated by the gradient-based function. Some of these designs were picked out and rerun through the objective function manually with their associated lambda value, and found to be valid Pareto designs. When the design is then plugged into the gradient-based function manually, the design is actually *worsened* by the optimizer, causing it to travel up or down the Pareto front. Usually the final result is an optimal design for another lambda value but not the one being used.

It is assumed that this behavior of raising the objective function is a problem lying in the code that has not been found yet. As a takeaway, both Pareto fronts (GA-only and GA+SQP) should be consulted when it is time to choose a design configuration. In the future, the code should be refined such that the dual-stage optimizer can output as many options as the GA alone.

6. MECHANICAL DESIGN

The mechanical design and prototyping of the cubesat frame was performed in order to support the concept that the optimized cubesat frame is effective and manufacturable. Extensive mechanical simulations are not taken on for this research. A survey of cubesat missions, such as "The first 100 Cubesats" [9], indicates that mechanical failure is not a common failure mode on cubesat missions. Figure 23 is adapted from a chart in this paper and offers a breakdown of the failure modes among the first 100 documented cubesat flights.

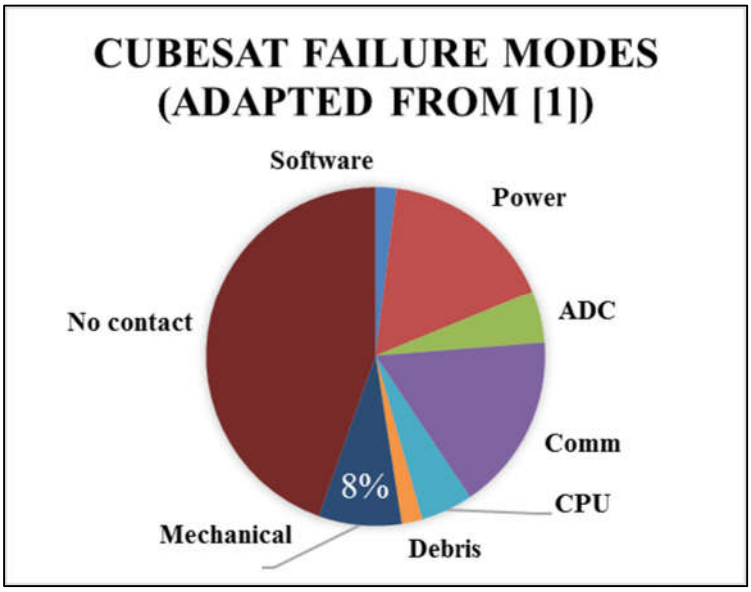


Figure 23. Cubesat Failure Modes

6.1 Main Chassis Components

The side panels and I members are designed to be cut using minimal processes on a CNC router. Figure 24 shows the sidepanel in isometric views. It can be cut using only 3 axes and 1 mill bit.

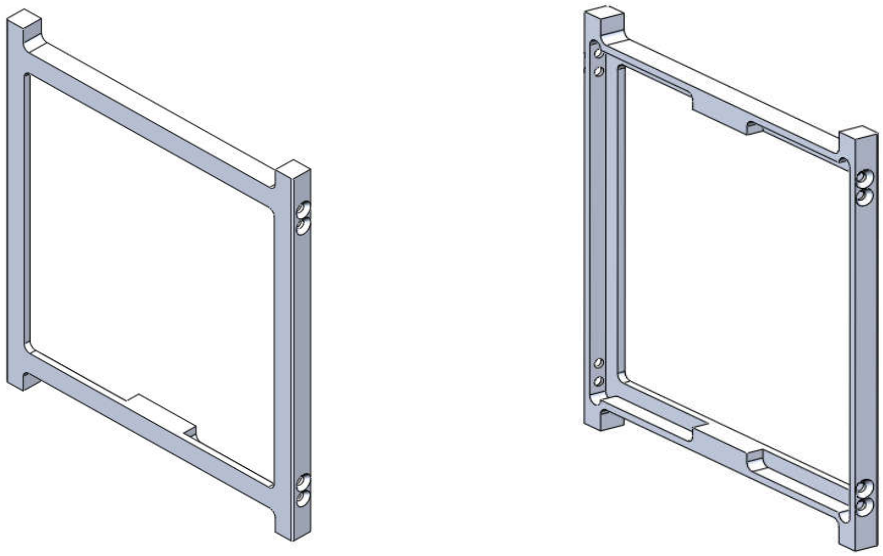


Figure 24. Sidepanel (SP) Front and Rear Iso Views

The holes in the sidepanel are clearance holes and mate with the threaded holes in the I-member. The I-member is pictured in Figure 25.

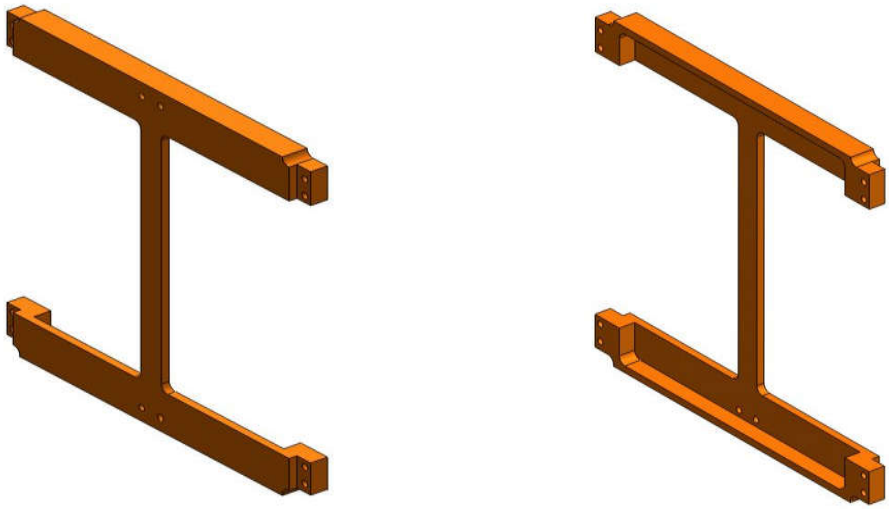


Figure 25. Imember Front and Rear Iso Views

There are two I-members per satellite. Truly, one piece was designed as two crossmembers but the web was added to allow the CNC mill to cut the piece as a whole. This simplifies the fixturing because it is easier to hold down one large member than to hold down two small pieces. It also adds rigidity to the frame. The

intent of the design is to keep the web as long as it does not interfere with the payload, and it can easily be relocated (or removed) in the future if it is necessary.

The fixed panel (FP) is the unique part of this cubesat design. It attaches to the outside of the frame and expands the volume to use the full interior of the PPOD launch apparatus. Figure 26 shows the fixed panel on both sides. The purpose of having the fixed panel not integrated with the other two components is to add modularity and make the machining process simpler.

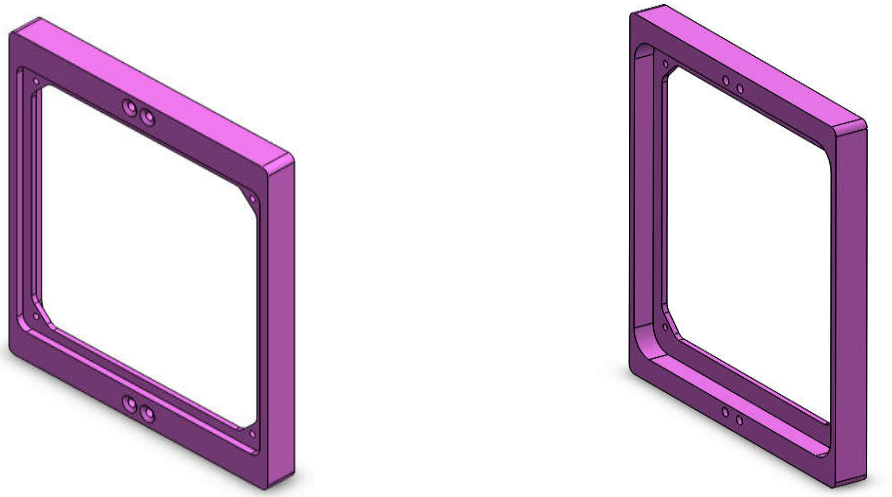


Figure 26. Fixed Panel Front and Rear Iso Views

The FP is designed as a platform to have multiple variations. Two basic variations are to have an inert cover sitting on the outside or to have a solar panel embedded in the outside, both of which fit readily. Another variation would be a fixed panel which contains a deployable solar panel array. With the FP being its own module, different FP's can be interchanged on cubesat main frame without any redesign of the main frame.

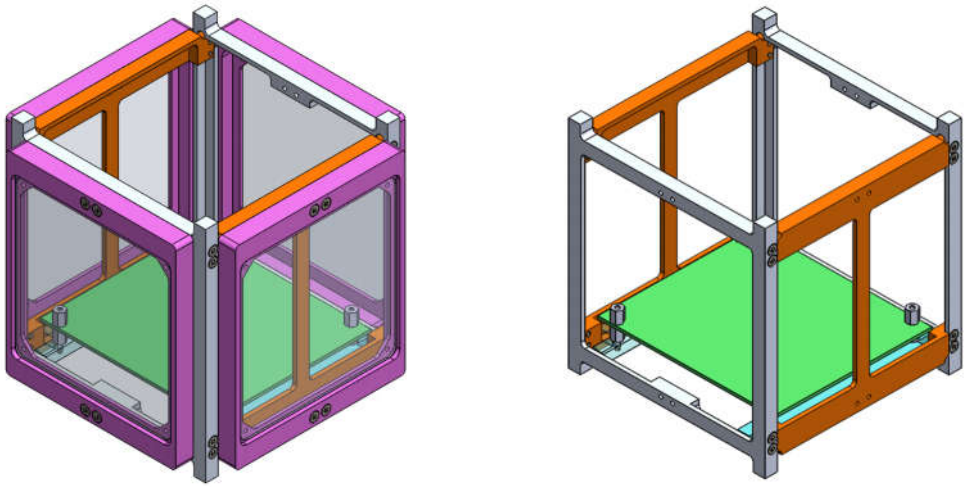


Figure 27. Cubesat Assembly

6.2 Fixed Panel Design

The Fixed Panel (FP) serves multiple purposes. It expands the chassis from a 100x100mm envelope by 9mm on each of the fours vertical sides and makes that space available internally for more payload. This panel, as shown in Figure 51, has a recessed shelf which holds either a flat panel (left side) or a solar cell (right side). A third variation of the FP contains a compact, deployable antenna. The conceptual design for this antenna is found in the mechanical design section.

6.3 Preliminary Circuit Board Mount

An extra feature was required accommodate part of a payload. The NanoRacks Embedded System Interface (NESI) board described in [17] is an example of a command and data handling (CD&H) unit that could operate the cubesat electronics. Therefore, it was integrated into the design for demonstration purposes. The NESI board, shown unpopulated in Figure 28, is a circuit that implements a PIC24 microcontroller and adapters to host several sensors and accessories. Advantages of the NESI board are that it fits nicely inside the cubesat envelope and that it has been space proven through operating multiple experiments on the international space station in the past.

In order to fix the NESI board inside of the cubesat, the adapter brackets shown in Figure 28 were designed and fabricated. Using standoffs as hardware allows further payload to be mounted above this PCB.

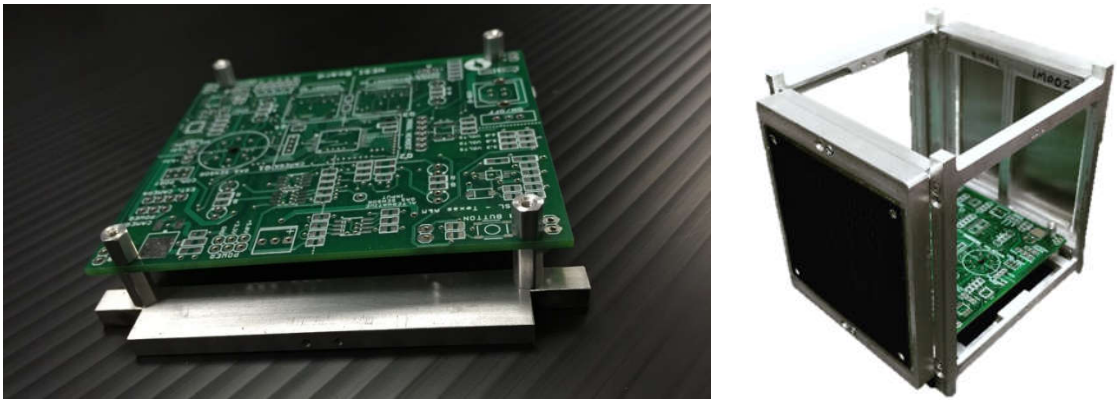


Figure 28. NESI Printed Circuit Board and Adapters

Integrating this board was the first step in designing the cubesat to carry a payload. Another likely candidate, aside from a customer-supplied CD&H unit, is the MISL stack. The MISL (Multiple Integrated Stackable Layers) is another space-qualified compact hardware architecture. This circuit suite, described in detail in [18] was also developed by the Texas A&M Electronic Systems Engineering Technology department in partnership with NASA-JSC.

6.4 Deployable Antenna Design

The antenna design concept aimed to meet the following criteria:

- Adjustable antenna length to accommodate different communication modules
- Design that is stowed inside the cube envelope until it is time to deploy
- Able to be deployed using only 1 IO signal and minimal amperage.
- Package such that maximum usable space remaining in the cubesat.
- Integrates into existing frame design
- Manufacturable with minimal moving parts

The resulting design meets all of the criteria except that a prototype has not actually been manufactured. Manufacturability is accounted for but has yet to be proven. The antenna itself consists of a guitar string manufactured by Ernie Ball. The composition is a non-wound, tin-plated tempered high carbon steel, according to the team at Ernie Ball, Inc. The wire appears circular but it has a hex shaped core. It is closest to ASTM material grade A228 Music Wire. The package is shown in Figure 29 (left).

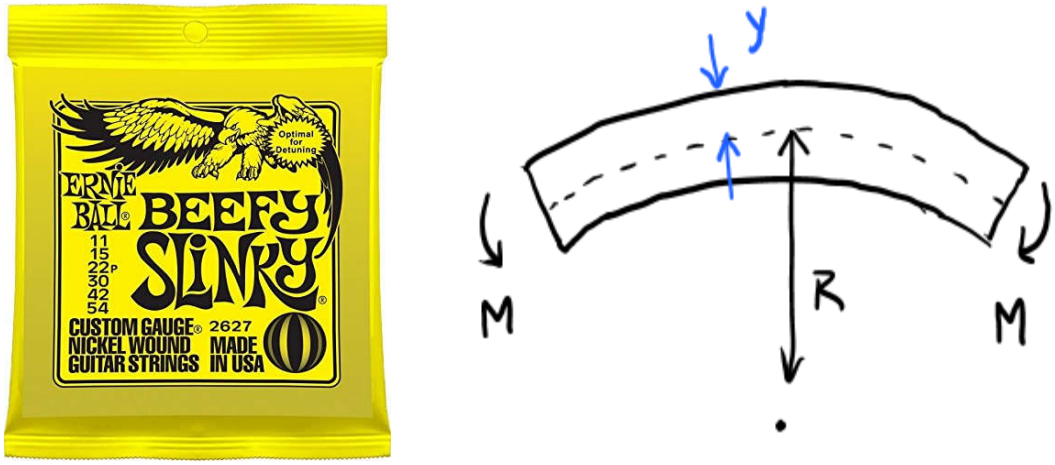


Figure 29. Music Wire for Antenna

The wire must be designed to be stowed in a coiled configuration but it must remain in elastic deformation. So, the minimum radius is calculated that the wire can be coiled such that it does not reach yield stress. Matweb.com shows the Yield Strength of this ASTM A228 material to be on the order of 363 ksi and the Modulus of Elasticity to be 30,000 ksi.

The maximum yield strain in a bending beam is shown in eq 6.1 where y is the distance from the neutral radius to the outer surface of the beam. R is the neutral radius, and sigma is the yield stress.

$$R = Ey/\sigma \tag{6.1}$$

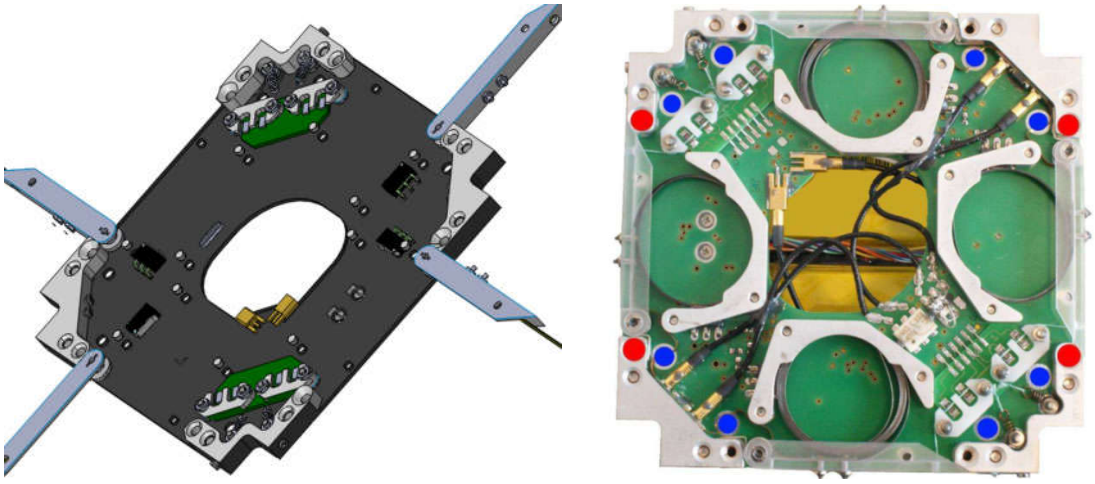


Figure 30. ISIS Deployable Antenna

For a wire with a radius of 11 gage wire, the minimum coil radius R is found to be 31mm. This means that it can fit safely within the cubesat fixed panel! The wire is able to be coiled inside of the fixed panel without the risk of plastic deformation. A CAD model in Figure 31 shows the antenna in a stowed configuration inside of the fixed panel. This design allows for the length of the antenna to be adjusted without any design change. Also, multiple antenna poles can be coiled inside the disc and released by just one actuator.

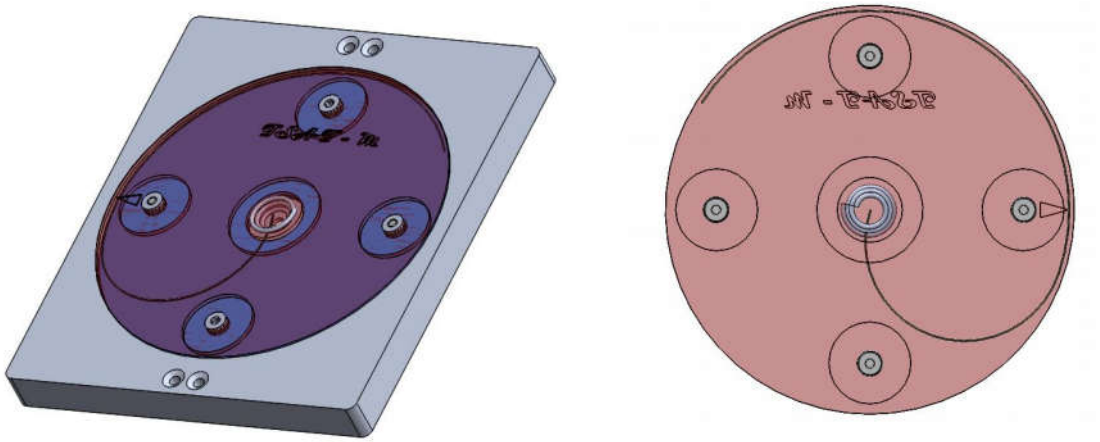


Figure 31. Coiled Antenna

In order to deploy the antenna, the disk that holds the wire would extend from a flush position to a protruding position. The motion of the disk is constrained by four shoulder screws which are fixed in the antenna disk and slide through holes in the fixed panel, shown on the left in Figure 32. Before deployment, the disk is held flush by a component that fills the gap. When this component is removed, the nested coil spring in the center will actuate the disk. Once the disk is protruding from the face of the satellite, the antennas poles are free to uncoil.

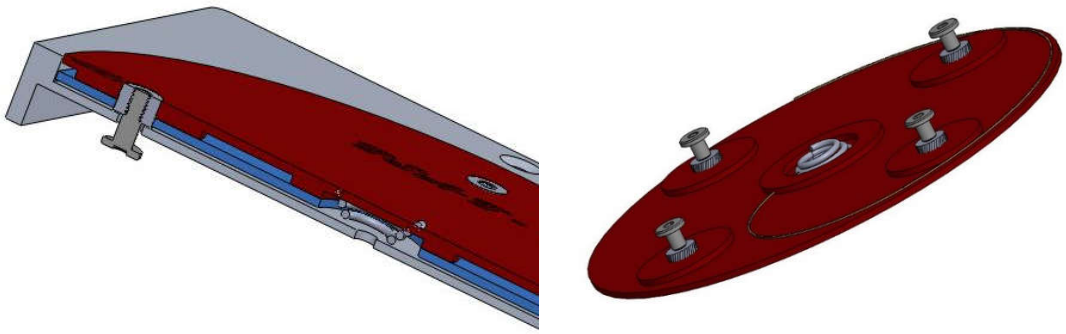


Figure 32. Antenna Deployment Mechanism

6.5 Finite Element Analysis & Simulation

For this design, a brief finite element analysis was constructed to form general observations on the frame strength. The study is not comprehensive enough to have confidence in the results without physical testing of the prototype. The choice to make only a brief study was influenced by other work that studied cubesat mission failures. "The flight history indicates that more time needs to be devoted to system-level functional testing rather than mechanical, thermal, and radiation issues." [9]. With this consideration the FEA process was made quick and efficient.

The FEA study begins by creating a mesh for the parts. The mesh is made from just the Sidepanel and the I-member in an assembly of 4 total parts. To take place of the fasteners (screws), a rigid connection was made between the hole in the Sidepanel and I-member (shown in blue and purple respectively in Figure 33 (left). This constrains those two faces to move together, which is a fair assumption since steel screws will be used. Only eight total fasteners were created in the assembly instead of the sixteen screws that are possible, which leads to a conservative study. The green arrows in the right hand of the figure show where the feet are constrained to the ground plane with one foot fixed in the x and y directions.

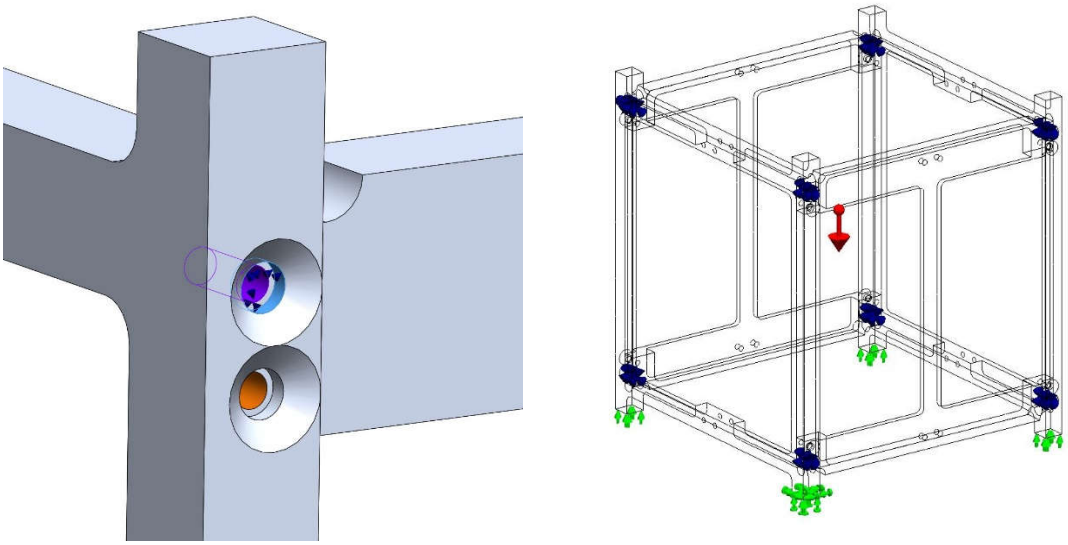


Figure 33. FEA Rigid Connections

Two different types of loads were analyzed on the structure. When the payload and external panels are fastened to the cubesat, these loads will be connected via the remaining holes in the structure. So, the first analysis involves bearing forces in the negative y-direction distributed equally among these holes. Figure 34 shows where the loads were located, with 50 Newtons applied to each pair of mounting holes for a total of 200N applied. The exaggerated deformation is shown in the right hand of the figure.

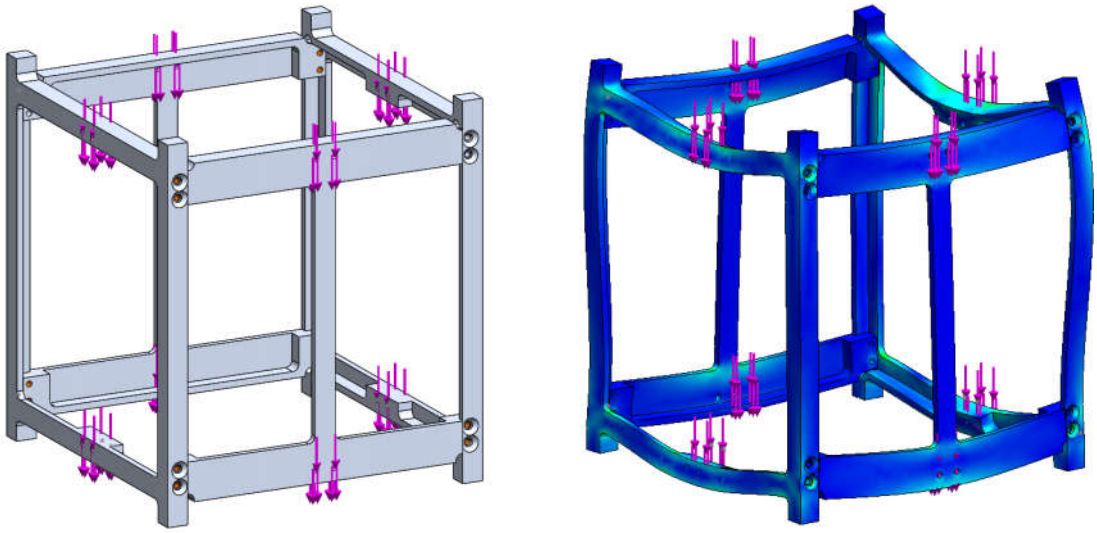


Figure 34. Cubesat Under Bearing Loads

This load translates to a 45lb load which yields a minimum factor of safety of 3.9 in all regions of the material. ISO clipping views are shown with regions in red falling below four (left side) and below fifteen (right side) in Figure 34. An interesting point from this study is that the thin wall of the I-member which is (only 2mm in thickness) does not appear to be one of the high stress areas even though it carries a load directly. This member was not originally designed to have a threaded hole and was therefore not reinforced in the same way that the square panel is reinforced. From this study, these holes may be selected as through-holes such that the member is clamped by a nut and bolt and the requirement for adding thickness may be eliminated.

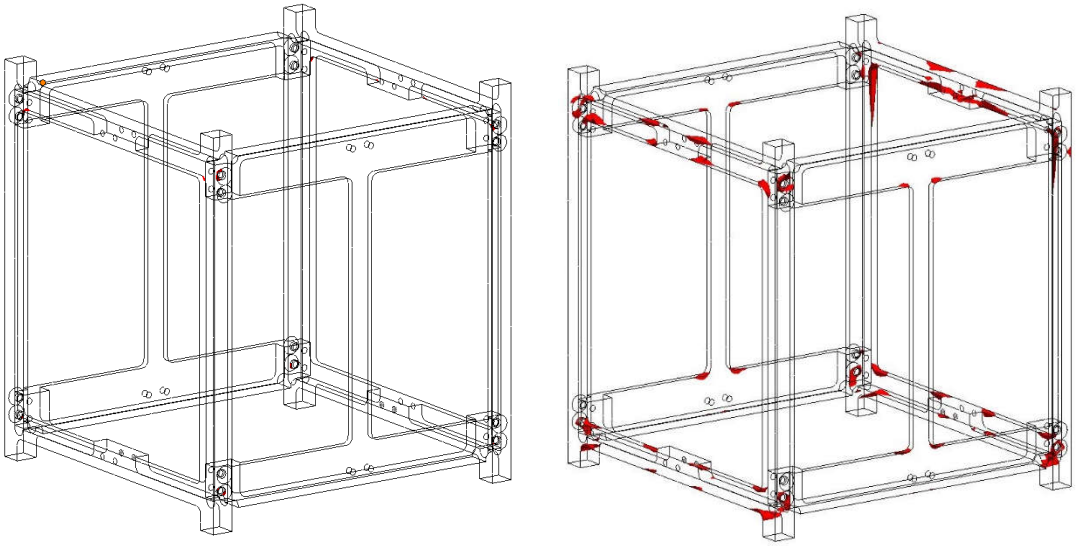


Figure 35. ISO Clipping Views under Bearing Loads

The second FEA study loads the satellite with a gravity force distributed evenly on the unit. This might be more representative of realistic loads because the designer should choose to relocate the fasteners to a place that creates the most even distribution of weight. Admittedly, the initial placement of the four fasteners on each side of the current design is a poor choice for attaching a Fixed Panel which is nearly the whole width of the cube.

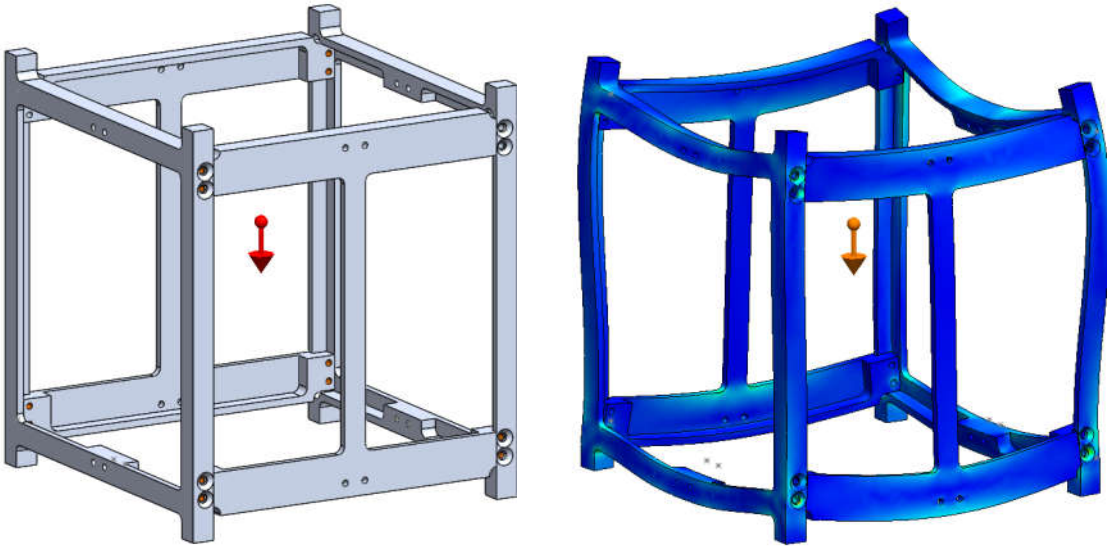


Figure 36. Cubesat under 1000g Gravity Load

For the 1000g gravity load, ISO clipping views are shown in Figure 37. The left hand side highlights regions with FOS below eight and the right hand side highlights FOS below fifteen.

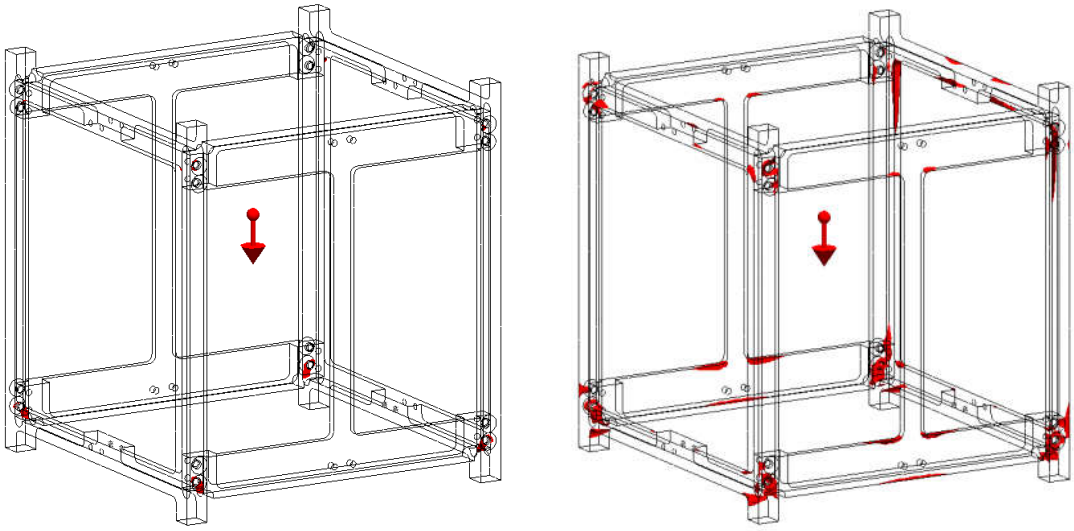


Figure 37. ISO Clipping Views under Gravity Load

To complete the Finite Element Analysis study, a convergence study was performed. Figure 38 shows meshes created with course and fine element sizes, both of which were used to populate the convergence data in Table 9 and Figure 40. Figure 39 shows the mesh settings for the initial gravity loading analysis along with a view of the mesh. A curvature-based mesh was used which increases the density of the mesh in more complex areas of the geometry.

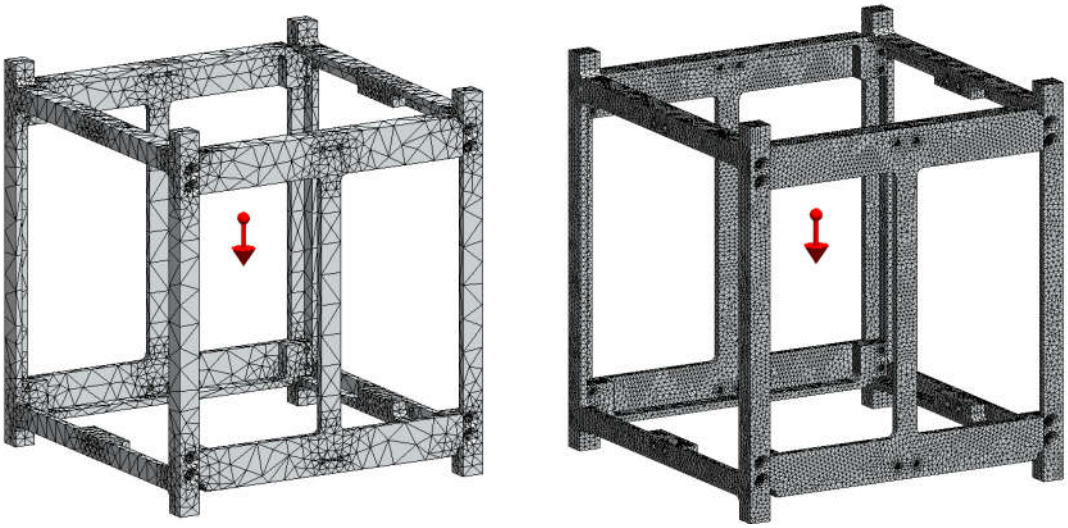


Figure 38. Mesh with Max Dimension of 8.64mm and 1.5mm

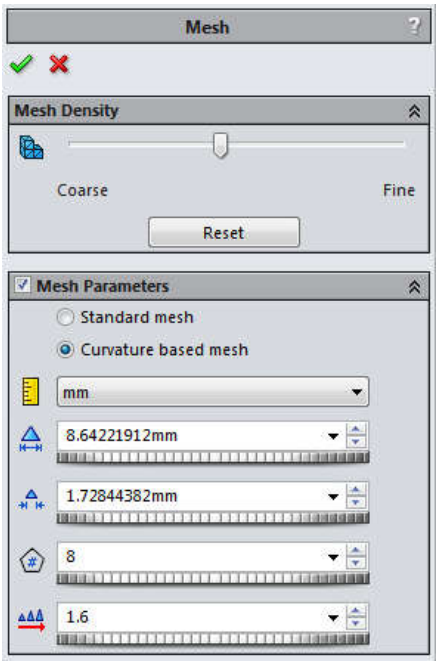


Figure 39. Mesh Settings

Table 9. Convergence Results

Max element dimension	Minimum FOS
11.8	3.23
10.6	3.21
8.65	3.19
7.66	3.29
6.78	3.57
6.08	3.28
5.49	2.92
4.91	2.93
4.32	3.35
3.93	2.40
3.00	3.16
2.50	3.11
2.00	2.48
1.50	3.09

The convergence study shows that results vary as much as 1.1 Factor of safety as the mesh size is changed and the analysis is rerun. The Factor of safety indicated refers to the region of maximum stress in the cube versus the yielding stress of the material (6061 T6 aluminum).

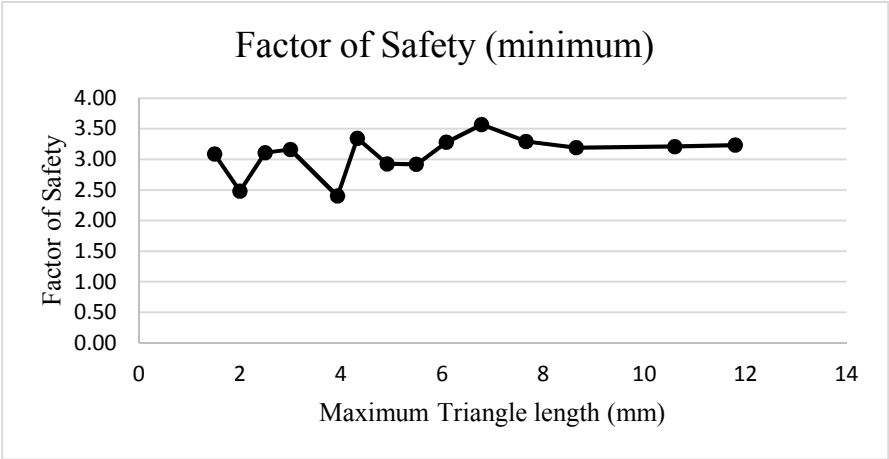


Figure 40. Convergence Study for Factor of Safety

This convergence is not as nice as we would like to see, but it has a relatively steady indication of 2.5 to 3.5 FOS. Were the FEA study was to be relied on alone with no prototype, it would be smart to consult a group who is more experienced in FEA to help draw conclusions.

If the FEA could be trusted fully to indicate the strength of the satellite, then it would be possible to design a frame which is less geometrically similar to a benchmark design and still trust that the frame is mechanically sound. For the existing study, the evaluation of the design is partly based on its similarity to a flight-tested unit.

7. PROTOTYPING

The process for machining the aluminum panels for the cubesat is as follows. The Solidworks parts are exported as STL files into Mastercam. In Mastercam the contours can be selected from the part, the origin is specified, and the working tool is specified. Mastercam generates a numerical control code for each toolpath that the user creates. The code, also known as G code, is then imported to the software called TECHNO-CNC which runs this particular CNC router. Figure 41 shows the router used in manufacturing the first prototypes.

Figure 41. Patriot CNC Router Used for Rapid Prototyping

7.1 Manufacturing Quotations

In the early stages of design, I sought quotes for the cost of prototyping. Two models were sent to the Texas A&M Wind Tunnel machine shop which performs machining jobs for the public [19]. First a model from the ISIS cubesat was quoted at 10 hours of machining time, shown in Figure 42, left. Then I created a simplified design with a resulting quote of only 3 hours.

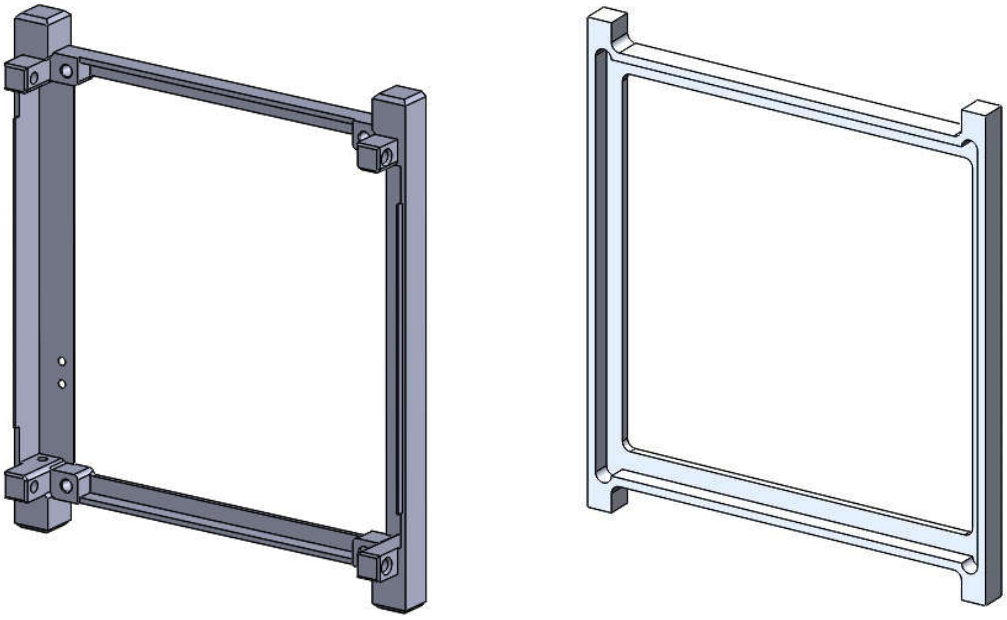


Figure 42. 3D Models for Machining Quotes (ISIS-Left Malawey-Right)

Several measures were taken to create a model that performs the same purpose but reduces the machining time. First, the chamfers were removed. These chamfers reduce sharp edges and help with safety during handling but there is no requirement for chamfered edges in the CDS. Secondly, radii were adjusted such that a larger cutting tool could be used. This allows the feed rate to increase for faster machining. It also eliminates the need for any tool changes because the whole part can be made with just one end-mill. Additionally, all features that require a third axis were removed. This means that a simpler machine can be used to make the part (a 3 axis mill instead of a 4-axis). However, it leaves the design with no attachment point for the mating part. To be realistic, the second quote should be increased to allow for re-fixturing and drilling at least 4 holes in the sides of the frame. Finally, the overall thickness of the part was reduced so that a smaller material stock could be used and less material is wasted.

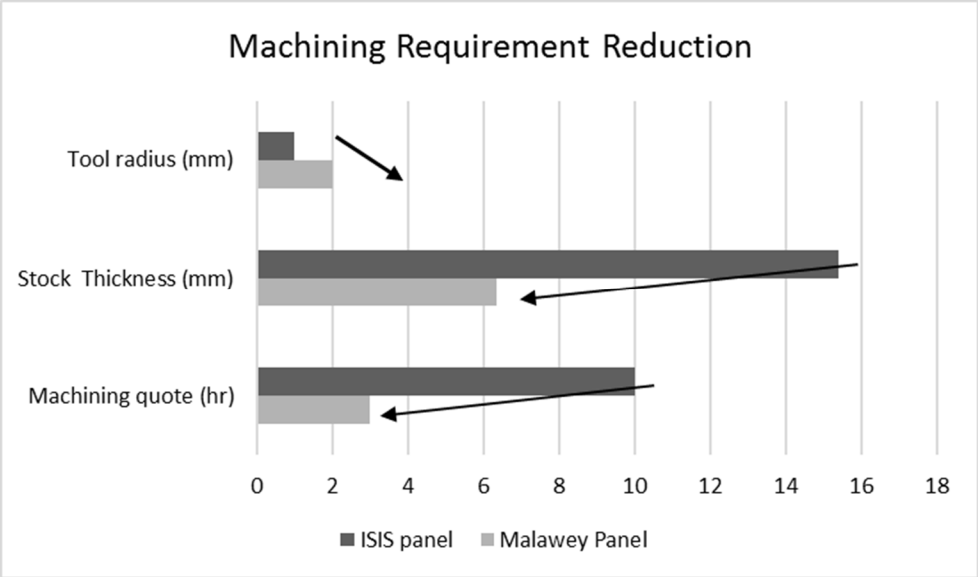


Figure 43. Machining Requirement Reduction

It is clear that some of the features in the ISIS panel serve a purpose other than adding complexity to the design. As my design moved forward I would find that a feature may need to be added in order to attach another part or meet a newfound criterion such as devoting one standoff to hold a "deployment switch." Because of this, my design cannot remain quite a simplistic as the one shown on the right in Figure 42. However, the same strategy (reduce the stock, increase the tool size, and reduce features) is taken throughout the whole design and leads to a lower production cost. Figure 43 shows the overall adjustments to the machining requirements that were made.

7.2 Fixture Design

To maximize the efficiency of the CNC program, we wanted to fulfill the following objectives

- Minimize setup time for placing the raw material
- Program the Mill to make multiple parts at once and take advantage of the mill size
- Minimize the number of tool changes

Since the main cube frame is made from two different pieces, the program is written to make both the Sidemember and the I-member from one piece of raw material. This allows:

- Only making 1 fixture design for both parts
- an opportunity to improve *either* part on *every* machining iteration.

The fixture designed to meet these requirements is shown in Figure 44. The first run did not result in perfect parts. Once the g-code and material setup process were tuned, the program was able to make one whole cube frame in just 2 runs.

Figure 44. CNC Fixture

The side holes could not be drilled in the H member by this CNC machine without designing additional fixturing. For this feature, the drawing was handed off to a campus machinist to drill and tap these holes.

7.3 CNC Program Development

Developing a Computer Numeric Control (CNC) program was a significant barrier to overcome before beginning the machining. All of the tool path design was created in MasterCam, which is featured in the following section.

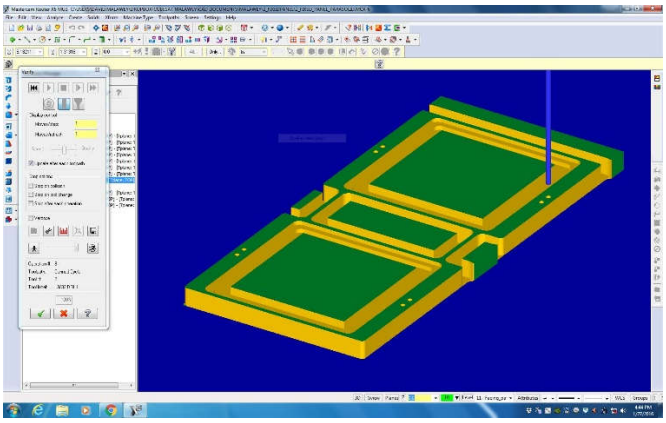


Figure 45. CNC Simulation

After the raw stock is set up in the computer as well as the tool specified, and the paths selected as well as the proper offsets added in (left hand, right hand, or none) the program is ready to output a graphical simulation of the workpiece. The workpiece for cutting two fixed panels is shown in Figure 45.

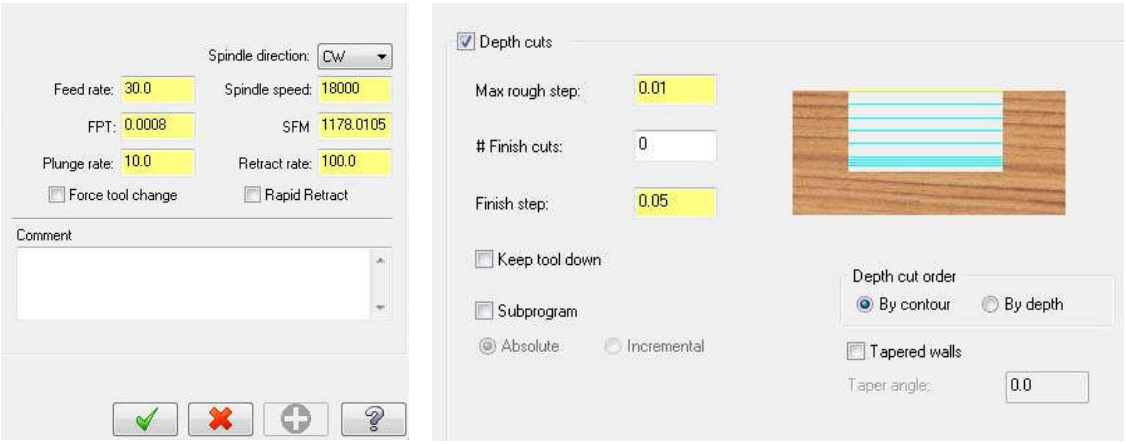


Figure 46. Speed and Feed Setup

The speeds and feed rates of the CNC tool were calculated by hand considering the tool diameter, number of flutes, and material properties. Figure 46 Shows these settings for the 1/8" end mill which was used for most operations. Conservative values were used at first, and then raised to speed up the machining. The right side of Figure 46 shows the program's settings for taking cuts at multiple depths. To achieve a deep machining path, many sweeps of the tool are taken at 0.01-inch depth increments. This machine speed is limited in comparison to most CNC's because it has no lubricating coolant.

Mastercam generates a toolpath model as shown in Figure 47 (left) after the contour is selected and settings are entered. Then Mastercam can generate G-code Figure 47 (right) which is actually used by the machining software.

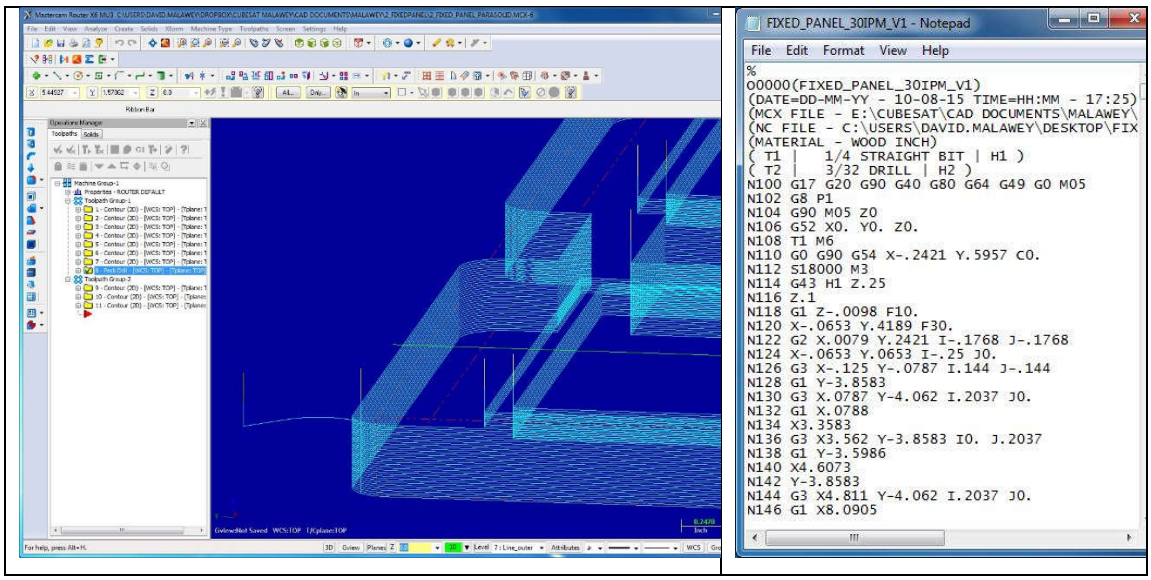


Figure 47. Toolpath Simulation & G-code

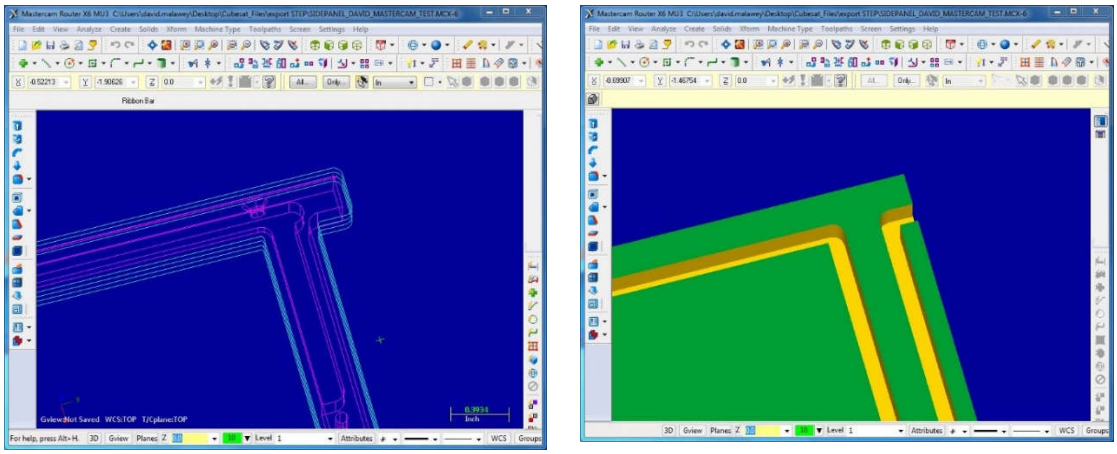


Figure 48. Design for Mill Tool

Special considerations have to be made in the design such that the satellite parts can mate after they have been cut. The right hand of Figure 48 shows a part that would not successfully mate at the inner corner, and the left hand part has been adjusted to create clearance for this mating part.

7.4 Prototyping Results

7.4.1 Machining Results

During the machining process, measures were taken each iteration of the parts to improve the process. The overall goal was to prove the parts can be manufactured using the expected machines while minimizing the time taken to produce the parts, and meet all required functionality.

Since the machine used to cut the parts is one of the most limited CNC machines (actually a router and not a mill), successful implementation using this machine suggests that speed and tolerances could be met by any common CNC mill.

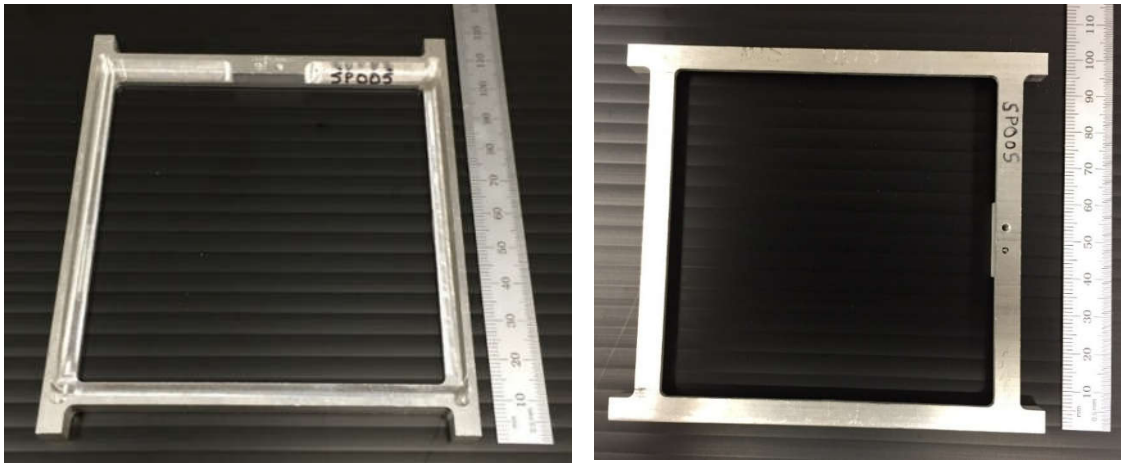


Figure 49. Machined Side Panel

The first member created was the side panel. The fifth iteration of the side panel is shown in Figure 49 before the holes were drilled on the sides for connection to the I-member. Details on this result follow later in this section.

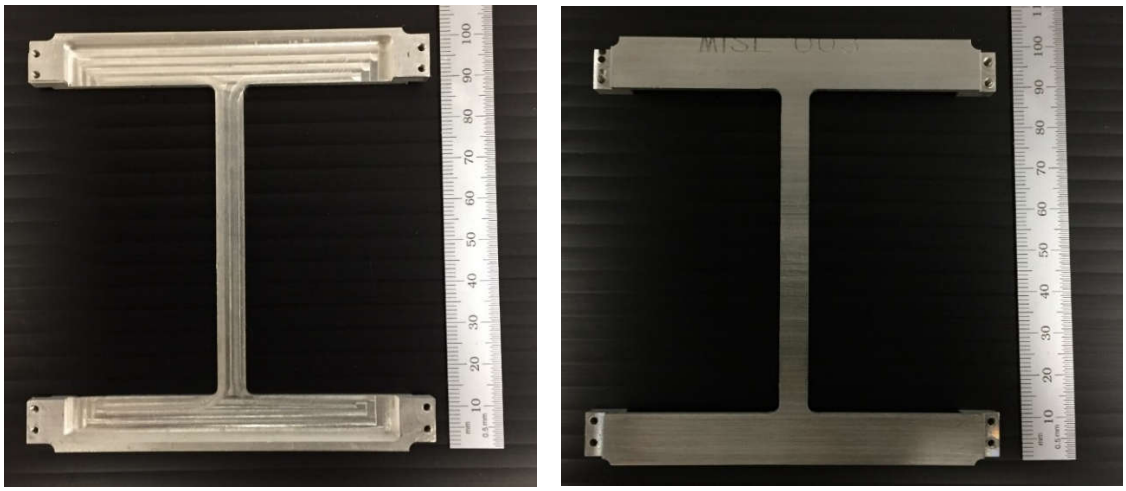


Figure 50. Machined I-Member

The second component that was machined was the Imember. One of the machined I-members is depicted in Figure 50 above. Details on the machining follow in this section.

Figure 51. Machined Fixed Panel with Plate or Solar Cell

The Fixed Panel (FP) was the last component to be machined and is shown in Figure 51. The FP prototype successfully uses both configurations and keeps all material within the specified envelope. The aluminum panel is held in place with thin strips of Kapton tape, and the attachment of the solar panel will require a special strategy that considers the electrical contact requirements.

Six iterations of each side panel and I-member were created and main dimensions were checked. These dimensions are taken from the drawings included in Appendix B. The results of these measurements are

shown in the graphs within Figure 52 and Figure 53. The solid horizontal lines in the results indicate a tolerance which, if not met, may cause interference with a mating part. The dashed horizontal line indicates a tolerance which, if not met, would not cause interference but may result in a part that does not meet design intent.

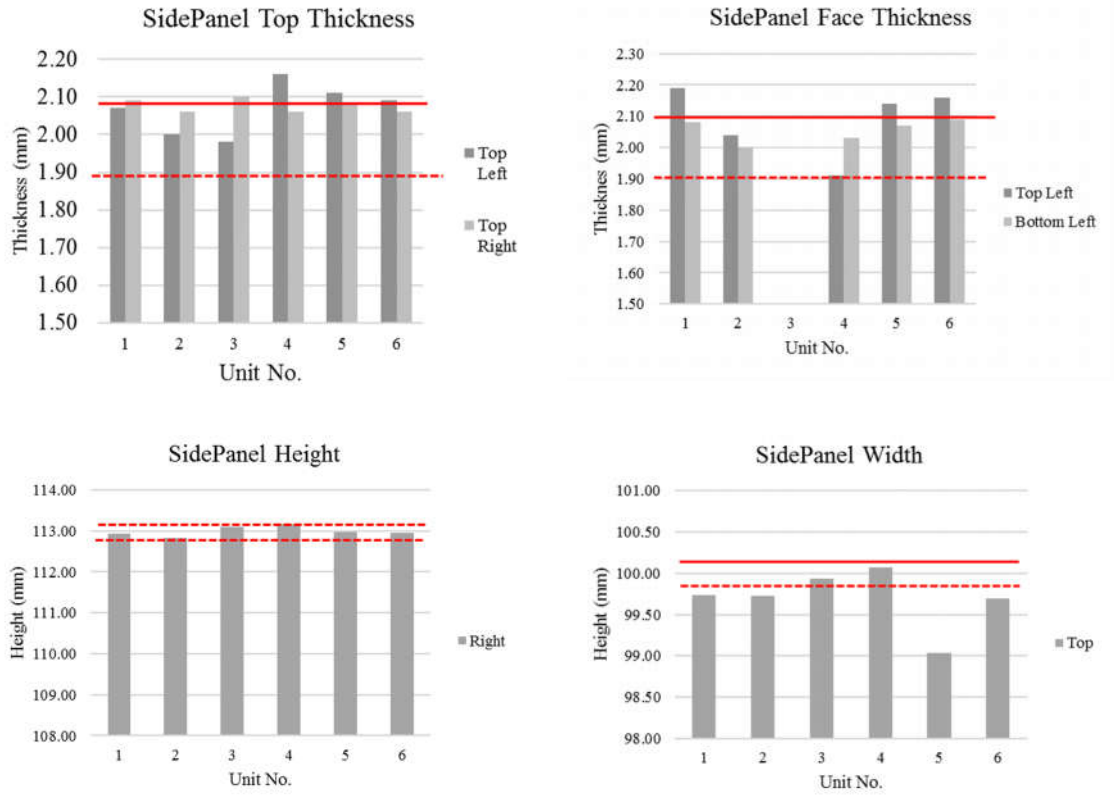


Figure 52. Sidepanel Measurement Results

There is not a trend of continuously improving tolerances with successive parts. Rather, an attempt was made to maintain or improve tolerances while also revising the CNC cutting program, adjusting the fixturing strategy. If the machining process were to be finalized, the best measurements met by these parts would arguably be a repeatable result.

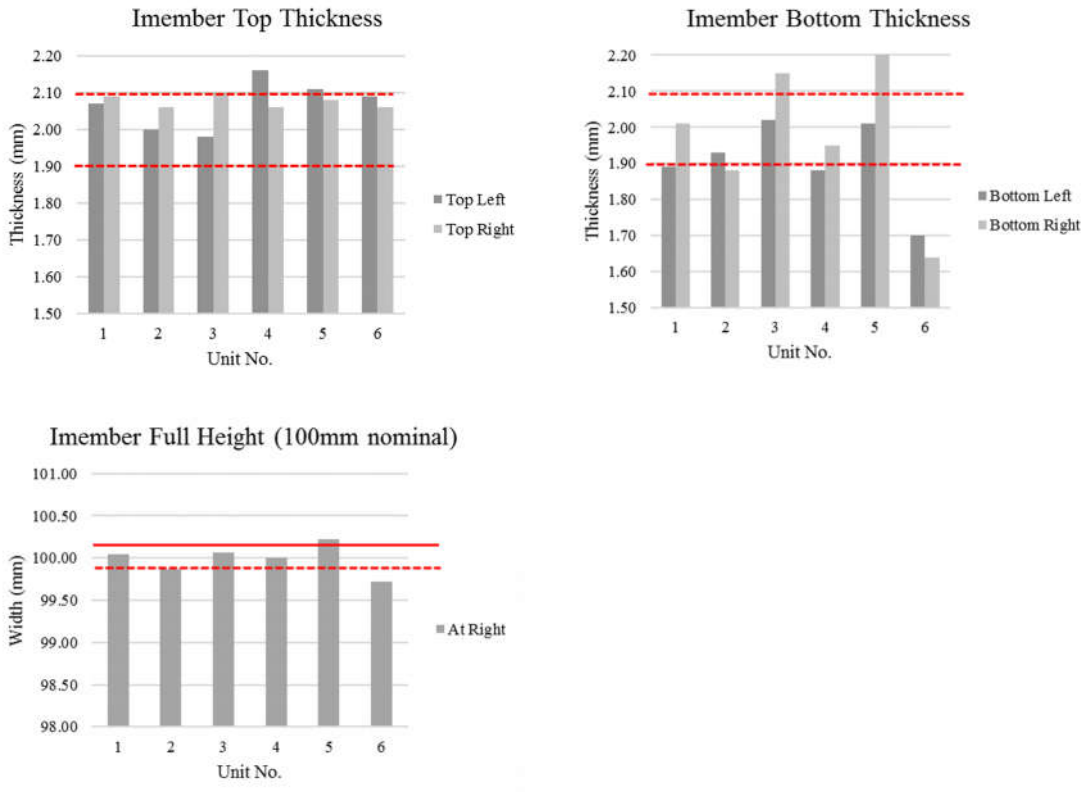


Figure 53. Imember Measurement Results

Even with continually changing machining processes, both components yielded some parts that met tolerances. This is taken as evidence that the machine, the tool, and the material in combination have the capability to produce good parts. The process of fixturing, material preparation, and tool changing should be refined in order to repeat the best tolerances resulting above.

7.4.2 Parts Quality Checks

The checking of quality on the manufactured parts extends beyond meeting tolerances. Observations were made on each part to determine if they would satisfy the intended design overall and if they had any features that indicated a non-repeatable manufacturing process. In other words, it would not be sufficient to have a manufactured part that meets a tolerance by luck rather than true manufacturing control.

The quality check on the parts offers notes which are summarized in Table 10 give information on the parts condition. Note that the NG the parts labeled "NG" for "no good" are still functional but had some characteristic that needs to be improved for the next iteration. This is not a problem but a characteristic of designing the manufacturing process simultaneously with creating first-off prototypes. Most of these parts are still meeting the critical (interference) tolerances, assemble-able, and would pass a strength test.

Table 10. Parts Quality Check

Part	Part Number	Condition after Machining	Result
Square Panel	SP001	Thicknesses are within 0.1mm. Hole for fixed panel is cut in the wrong place. Additional hole was cut for correction.	NG (no good)
	SP002	0.2mm low on overall width	OK
	SP003	Face pocket is cut too deep, leaving only 1mm thickness (-1.00mm from nominal).	
	SP004	Imperfect alignment along top edge	NG
	SP005	All measurements within 0.2mm. Tap is broken off in threads.	NG
	SP006	One rail is 0.4mm thinner than nominal thickness. NG	NG
I-Member	IM001	Meets tolerances	OK
	IM002	Top thickness is (-) 0.24mm from nominal. Gap at bottom mating location is too large.	NG
	IM003	Thicknesses are correct but gap is too large (approx. 0.8mm) at one side, against mating piece.	OK
	IM004	Rail is as thin as -0.14mm from tolerance.	NG
	IM005	Top rail is 0.37mm thinner than nominal.	OK
	IM006	Tap is broken off inside one of the threaded holes.	NG
Fixed Panel	FP001	Mounting holes are not aligned. Inside pocket has rough machining due to insufficient clamping	NG
	FP002	Improved but still insufficient clamping: rough machining	NG
	FP003	Improved but still insufficient clamping: rough machining	NG

The fixed panel quality checks yielded multiple iterations of process improvement. At the end of the cutting process, small slivers of material connected the panel bodies to the scrap material which was clamped. These tabs were designed to give some rigidity to the part during the end of the milling process, and to be removed after milling. These tabs were increased in thickness to reduce vibrations of the material which caused rough cutting.

A future addition to this manufacturing process would be to add a feature to the original mounting fixture which creates a fast way to add an additional clamp to this material to supplement the job of the tabs.

7.4.3 Process Improvement

Throughout the manufacturing process, several steps were taken to both achieve the desired tolerances and reduce the time required to machine the parts. Figure 54 summarizes the steps taken to reduce machining time and the results.

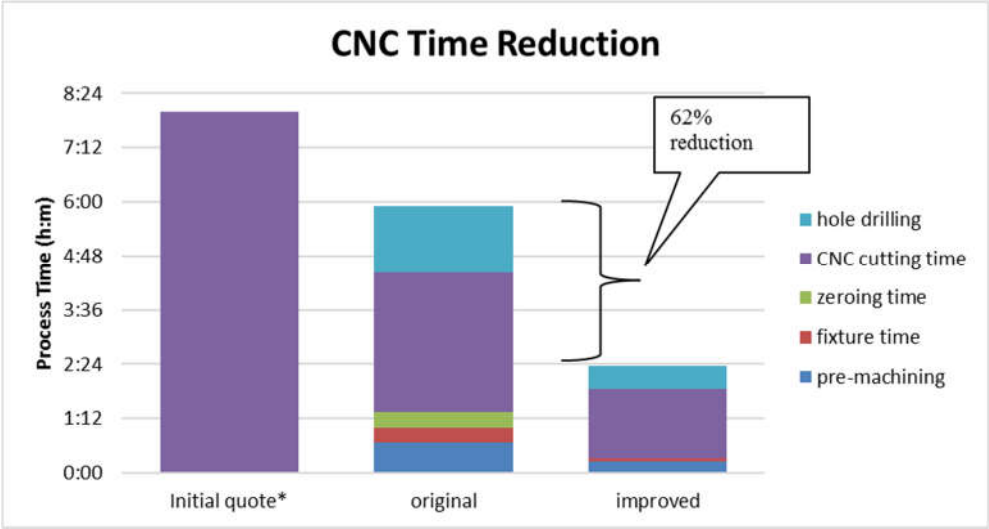


Figure 54. Manufacturing Time Reduction

The process time represents the time it takes to cut out two copies of the side panel, both of which are used to build a 1U chassis. The initial quote refers to the three-hour estimate from Texas A&M Wind Tunnel Machine Shop with an additional hour added for drilling holes, totaling eight hours for two pieces. The original process time refers to the machining time required during the first attempt at building the parts. The measures taken to reduce this timing are described in Table 11.

Table 11. Manufacturing Time Breakdown

Step	Original Process		Improved Process	
pre-machining	0:40	finish all 4 sides x2 panels. 2 rough cuts, 8 finish cuts	0:15	1 rough cut, 2 finish cuts, 4 rough holes drilled
fixture time	0:20	align parts in CNC (2x), use 2 sided tape	0:05	(alignment is done by fixture), use 2-sided tape
zeroing time	0:20	zero the tool against the parts (2x)	0:00	(tool already set at zero)
CNC cutting time	3:05	cut contours using 1/8 flat end mill	1:31	cut contours with 1/4 endmill, cut small radii with 1/8 endmill
hole drilling	1:30	machinist drill all 18 holes, taps done by hand	0:30	CNC drills 10 holes, 8 by machinist, taps done by hand
Total Time	5:55		2:21	

More than five hours were removed from the original quotation and more than three hours were reduced from the first machining iteration. This result, in combination with the notion that the machine was a minimal required machine to perform the job, suggests that time can be shed from existing designs for manufacturability.

7.4.4 Fitment Testing Results

The latest copy of the prototype was brought to NanoRacks LLC in Webster, TX. The staff at NanoRacks used their check fixture to insert the Cubesat (including fixed panels) and verify that it clears the envelope. Figure 55 shows the cubesat's successful fitment.

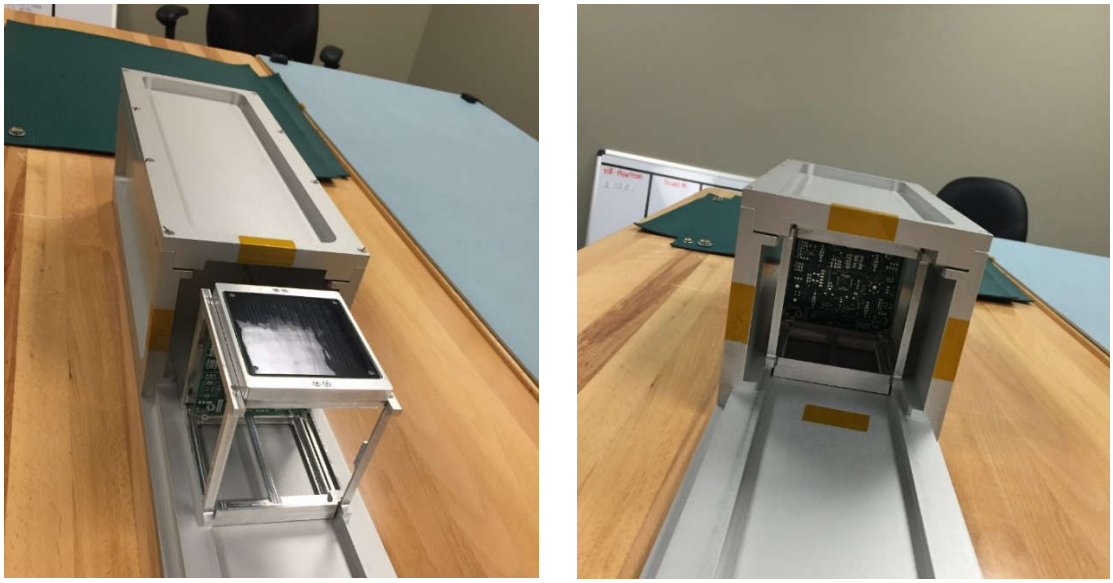


Figure 55. P-POD Fitment

8. CONCLUSIONS

8.1 Conclusions of this Research

The results presented in this thesis suggest several pieces of information to take away. First of all, in the optimization of a cubesat that is populated with heuristic data will inevitably include some discrete design variables and these can be effectively handled by a heuristic optimization method. Specifically, the genetic algorithm was explored and found to give an optimal result that is repeatable within a small variation using at most a few iterations of the optimizer. A few iterations for this case is acceptable because the low-fidelity model is able to run in less than one minute.

Furthermore, the heuristic method is enhanced by adding in a second stage of optimization using a gradient based method. The gradient method (in this case, SQP) makes the result repeatable to a fraction of a percent in the objective function. One drawback of the gradient function is that it causes some Pareto designs to be hidden when it comes to dual-objective optimization. It is expected that this issue could be resolved with further investigation.

Development of graphical outputs and sensitivity studies help to protect against impractical design decisions or violation of constraints that are not captured by the optimizer. These items help spur logical checks such as consideration of whether the aggregate volume of all components would fit in the satellite, regardless of mass of these components. The ability to successfully make these checks depends on correct study results coupled with knowledge of manufacturing and the multidisciplinary systems. Unfortunately the optimizer still cannot take the place of a design engineer.

On the facet of manufacturability, all evidence suggests that the cost of manufacturing can be reduced from the status quo. That is, satellite components similar in function to existing designs can be developed with a closer focus on manufacturability from the start of the design to achieve lower cost. This stipulates that design must be intended for one specific type of manufacturing process as it is created. In this case, the design was intended for CNC machining from bar stock. If the same design were to be attempted using a new method, it may no longer be optimal and could result in a higher cost than benchmarked components.

Most identifiable shortcoming in the estimates of manufacturability and cost in this study lie in the fact that the whole satellite was not created and flown. A small amount of uncertainty lies in additional changes that would be required to accommodate a full payload. Are there unexpected holes and features that need to be added? Would they impact cost or strength? These kind of uncertainties also could be asked regarding

successful vibration and temperature testing, etc. However, the methodology aimed for conservative estimates on these areas to give correct overall conclusions on each item.

8.2 Future Work

Future developments on this work could be broken into two categories: an extension of the design into full implementation, or achievement of more depth in an area of the study.

Beginning with depth related topics, more heuristic data should be added to the optimizer model. Cubesats are a field that is growing exponentially and a vast body of data on subsystems is being produced. Even without detailed physical models, this kind of optimizer has the potential to offer highly insightful results that are detailed enough to truly build prototypes from. Components need to be fully understood before populating into the design, and any special requirements of these commercial components need to be described in further constraint functions. An example of this kind of addition might be adding a list of magnetorquers to the model. These may add an additional power demand, but they may eliminate the requirement for cold gas storage and end up being an optimal design choice even at high costs.

A second depth-adding technique which also serves as a safety guard from unexpected failure would be to add multiple timescales. Demands for power and propulsion are met from a whole-mission point of view in this study but in reality they need to be met from a day-to-day basis and an hour-to-hour basis. The batteries need to be sufficient to power all subsystems while the satellite is in the earth's shadow, and the propulsion system needs to ensure it does not lock up due to extended thrust periods which drop the temperature of the cold gas containers. The smaller timescale constraints would be a large step in making the solver more realistic.

In the category of design extension, the first future job is to build and implement a satellite based on one of the optimizer's results. This process would cause new issues to surface so that the optimizer can be refined. Multiple implementations in series with improvements of the optimizer would be very valuable. The software can perform genetic optimization on the satellite while the engineers perform genetic optimization on the software, essentially.

The other concept that would make this design more complete is to enhance the coupling of the manufacturing process and the optimization process. If the CNC program could be generated as a module of the optimizer, the structure could become truly optimized for both cost and machining.

NOMENCLATURE

Abbreviations

Accy	Accessory
Batt	battery
C&DH	Command and Data Handling
CNC	Computer Numerical Control
COTS	Commercial off-the-shelf
FOS	Factor of Safety
FP	Fixed Panel
GA	Genetic Algorithm
IM	I-member
IO	Input/Output
MISL	Modular Integrated Stackable Layers
NASA	National Aeronautics and Space Administration
NESI	NanoRacks Embedded System Interface
NRCSD	NanoRacks Cubesat Deployer
Qty	Quantity
SP	Side Panel
SQP	Sequential Quadratic Programming

Symbols

E	Modulus of Elasticity
$g(X,P)$	Constraint function
I	Area Moment of Inertia
I_{sp}	Specific Impulse
$J(X,P)$	Objective Function
X^*	Optimal design vector at hand
σ	stress

REFERENCES

[1] "CubeSat Design Specification (CDS)," California Polytechnic, San Luis Obispo, 2009.

[2] L. NanoRacks, "NanoRacks CubeSat Deployer (NRCSD) Interface Control Document," Houston, TX, 2013.

[3] S. E. Fredrickson, S. Duran and J. D. Mitchell, "Mini AERCam Inspection Robot for Human Space Missions".

[4] A. Anis, "Cold Gas Propulsion System - An Ideal Choice for Remote Sensing Small Satellites," NED University of Engineering and Technology, Pakistan.

[5] S. A. Rawashdeh, "Passive Attitude Stabilization for Small Satellites," University of Kentucky, Lexington, KY, 2010.

[6] H. C. Kjellberg, "Design of a Cubesat Guidance, Navigation, and Control Module," University of Texas at Austin, 2011.

[7] J. L. D. C. J. M. J. Hwang, "Large-Scale MDO of a Small Satellite Usinga Novel Framework for the Solution of Coupled Systems and their Derivatives," in *54th AIAA Structures, Structural Dynamics, and Materials Conference*, Boston, MA, 2013.

[8] L. Y. Dae and C. W. James, "Design Optimization of a Solar Panel Angle and its Application to Cubesat 'CADRE'," University of Michigan.

[9] M. Swartwout, "The First One Hundred Cubesats: A Statistical Look," *Journal of Small Satellites,* pp. 213-233, 2013.

[10] P. Y. Papalambros and D. J. Wilde, Principles of Optimal Design, Melbourne, Australia: Cambridge University Press, 1988.

[11] Clyde Space Ltd., "CubeSat Solar Panels," 2005. [Online]. Available: http://www.clyde-space.com/. [Accessed 20 March 2016].

[12] Spectrolab Inc, "Spectrolab Photovoltaics," [Online]. Available: http://www.spectrolab.com/faqs-space.htm. [Accessed 22 March 2016].

[13] Juergen and Mueller, "Thruster Options for Microspacecraft: A Review and Evaluation of Existing Hardware and Emerging Technologies," *AIAA,* 1997.

[14] Marotta, "marotta.com," [Online]. Available: http://marotta.com/cold-gas-microthruster. [Accessed 10 February 2016].

[15] Quaker Chemical Corporation, "quakerchem.com," 2013. [Online]. Available: http://www.quakerchem.com/wp-content/uploads/pdf/skill_builders/no10_machinability_ratings.pdf. [Accessed 11 05 2015].

[16] D. Hsu, 30 4 2015. [Online]. Available: http://www.chemicool.com/.

[17] D. Karrer and C. Schimank, "The Nanoracks Embedded Systems Interface Board: An Innovative STEM Outreach Tool," in *ASEE Gulf-Southwest Annual Conference*, San Antonio, 2015.

[18] J. A. Morgan, "Modular Integrated Stackable Layer (MISL): An Academic-Public Sector Partnership for Rapid Prototype Development," in *ASEE Annual Conference and Exposition*, Seattle, WA, 2015.

[19] Texas A&M University, "Low Speed Wind Tunnel," [Online]. Available: http://lswt.tamu.edu/. [Accessed 10 February 2016].

APPENDIX A – MATLAB CODE

```
% optim_2stage.m calls the objective function into the optimizer
%-----------------------------------------------------------

clear all

%----------DESIGN VECTOR-------------------------------------
%(1)propellant (2)thruster type (3)structure material
%(4)#panels (5)#batteries (6) struct rail width

%----------GLOBAL PARAMETERS---------------------------------
P101=30; %mission duration, (days)
P102=0.33; %cubesat accy mass (kg)
P103=0.1; %solar panel sunlight(ratio)
P104=4; % default X1 (Nitrogen=4)
P105=3; % default X2 (Lee=3)
P106=1; % default X3 (6061=1)
P107=3400; %battery capacity (mAh)
P108=2.1; % power of 1 solar panel (W)
P109=0.5; %rate, disturbance per day (m/s/day)
P110=16.34; %prop tank volume (ml)
P111=2; % power factor of safety (FOS)
P112=0.1579; % min bending stiffness (m^4*GPa)
global P %create a global set of parameters
P=[P101 P102 P103 P104 P105 P106 P107 P108,...
    P109 P110 P111 P112];
clear P101 P102 P103 P104 P105 P106 P107 P108...
    P109 P110 P111 P112;

%----------GENETIC ALGORITHM APPROACH------------------------
LB=[1,  1,  1, 0,  1, .003];
UB=[9,  4,  3, 4, 10, .020];
x0=[1,  1,  1, 1,  1, .010];
tic
options= gaoptimset('Display','off','Generations',160,...
    'Elitecount',4,'PopulationSize',30,'TolCon',0.01,...
   'PlotFcns',@gaplotbestf);
[X]=ga(@(X) objective(X),6,...
```

```
    [],[],[],[],LB,UB,@(X) constraint(X),[1 2 3 4 5],options)
% "6" says that there are 6 design variables
% [1 2 3 4 5] calls out that these are integer variables
P(4)=X(1);
P(5)=X(2);
P(6)=X(3);
clear LB UB x0 time

%----------FMINCON APPROACH-----------------------------------
% for fmincon the problem is reduced to X(4,5&6) as
% (1)#panels (2)#batteries (3) struct rail width
LB=[0,  1, .003];
UB=[4, 10, .020];
x0=[X(4),X(5),X(6)]; %(use values from GA)
%x0=[4,  2,.008]; %(use values for study purpose)
% objective.m and constraint.m operate just on X(4,5,6)
% when length(X)==3.
options= optimoptions(@fmincon,'Display','off',...
    'MaxIter',150,'PlotFcns',@optimplotfval,...
    'Algorithm','sqp');
[X,FVAL, OUTPUT] = fmincon(@(X) objective(X),x0,...
    [],[],[],[],LB,UB,@(X) constraint(X),options);
%re-declare fixed parts of X to get full vector
time=toc;
%reassemble X-star from both optimizers
fixed=[P(4:6)]; %populate discrete variables from global P
vari=[X]; %populate the continuous variable
X=[fixed,vari]; %recompile X*

%-------------recalculate all vars with final X--------------
[j,s,p,b,y,c] = objective(X); %for regular evaluation

% %%
% %----------PARETO FRONT WITH GA AND SQP---------------------
%
%    % 'PlotFcns',@gaplotbestf);
% N=20; % number of iterations
% for i=1:N;
%     lam=i*(1/N);
% clear X
% % run GA
```

```
%     LB=[1,  1,  1, 0,  1, .003];
%     UB=[9,  3,  3, 4, 10, .020];
%     x0=[1,  1,  1, 1,  1, .010];
%     options= gaoptimset('Display','off','Generations',160,...
%         'Elitecount',4,'PopulationSize',30,'TolCon',0.01);
%     [X]=ga(@(X) dual_obj(X,lam),6,...
%     [],[],[],[],LB,UB,@(X) constraint(X),[1 2 3],options);
%     P(4)=X(1);
%     P(5)=X(2);
%     P(6)=X(3);
%     clear LB UB x0
%     [j,s,p,b,y,c] = dual_obj(X,lam);
%     cdata1(i,1)=c(3,1); % cost (g)
%     mdata1(i,1)=c(3,2)*1000; %mass (g)
%     solution1(i,:)=X; %design vectors
% % run SQP
%     LB=[0,  1, .003];
%     UB=[4, 10, .020];
%     x0=[X(4),X(5),X(6)]; %(use values from GA)
%     options= optimoptions(@fmincon,'Display','off',...
%         'MaxIter',150,'Algorithm','sqp','TolCon',0.01);
%     [X,FVAL, OUTPUT] = fmincon(@(X) dual_obj(X,lam),x0,...
%     [],[],[],[],LB,UB,@(X) constraint(X),options);
%     fixed=P(4:6);
%     vari=X;
%     X=[fixed,vari];
% % recalculate the output
% [j,s,p,b,y,c] = dual_obj(X,lam);
% cdata(i,1)=c(3,1); % cost (g)
% mdata(i,1)=c(3,2)*1000; %mass (g)
% solution(i,:)=X; %design vectors
% J(i,1)=j;
% end
% % "6" says that there are 6 design variables
% % [1 2 3 4 5] calls out that these are integer variables
% figure
% plot(cdata,mdata,'o',cdata1,mdata1,'x');
% xlabel('cost ($)');
% ylabel('mass (g)');
% title('pareto front');
% %
```

```
% %-------------tables entries---------------------------------
%propellant
ptable=...
 {'Hydrogen','Helium','Neon','Nitrogen','Argon',...
 'Krypton','Xenon','Freon','Methane','ammonia'};
%thrusters
ttable={'Moog 1W','Marotta .09W','Busek .89W','Lee .04W', };
%materials
mtable={'Al 6061 T6','titanium','A36 steel'};

%---------------create output UI-----------------------------
fig1=figure('Name','Results'); %give the fig a name and variable
set(fig1,'Position',[50 300 700 400]); %bottom,left,width,height
piedata=[p(3,4),s(3,1),...
    b(3,4),P(2)]*1000;
labels = {'prop','structure','solar + batt','accy'};
subplot(1,2,1);
pie(piedata,labels)
title('Total Cubesat Mass');

% Create the column and row names in cell arrays
massdata=[int32(1000*(p(3,4)))...
    int32(1000*(s(3,1))),...
    int32(1000*(b(3,4))),...
    int32(1000*P(2))];
rownames = {'mass (g)'};
% Create lower UI table
t = uitable(...
            'Data',massdata,...
            'ColumnName',labels,...
            'RowName',rownames,...
            'Position',[300 15 390 40]); %left,bottom,width,height
%create mid UI table
rowlabel = {'algorithm run time (s)','prop tanks (qty)',...
    'propellant mass (g)','pwr consumption (Wh)',...
    'solar pwr produced(Wh)'};
columnnames= {'Value'};
% if ~exist(time)
%     time=1;
% end
```

```
time=1;
    xdata2={time;p(3,7);int32(p(3,5)*1000);...
        int32(b(3,1));int32(b(3,5))};
    % xdata(1,2)= {'ptable(X(1))'};
t = uitable(...
            'Data',xdata2,...
            'ColumnName',columnnames,...
            'RowName',rowlabel,...
            'Position',[350 65 325 115]); %left,bottom,width,height
txt = uicontrol('Style','text',...
        'Position',[375 185 120 20],...
        'String','Table B: Relevant Data');
%create upper UI table
designvars = {'total mass (g)','propellant','thruster',...
    'material','solar panels','batteries',...
    'structure width(mm)','cost($)'};
columnnames= {'Value'};
xdata={int16(c(3,2)*1000);ptable{X(1)};ttable{X(2)};...
    mtable{X(3)};X(4);X(5);X(6)*1000;int16(c(3,1))};
    % xdata(1,2)= {'ptable(X(1))'};
t = uitable(...
            'Data',xdata,...
            'ColumnName',columnnames,...
            'RowName',designvars,...
            'Position',[350 210 290 175]); %left,bottom,width,height
txt = uicontrol('Style','text',...
        'Position',[375 370 120 20],...
        'String','Table A: Design Vector');
clear massdata mtable piedata ptable rowlabel rownames...
    ttable txt xdata xdata2 time columnnames designvars...
    options;
clear LB UB x0 OUTPUT fixed vari;

% %---------------cost output UI-----------------------------
fig2=figure('Name','Cost'); %give the fig a name and variable
set(fig2,'Position',[50 300 400 400]); %bottom,left,width,height
% batt+solar+prop+thruster+material+machining
piedata=[c(2,1),c(2,2),c(2,3),c(2,4),c(2,5),c(2,6)];
labels = {'batt','solar','propellant','thruster',...
    'material','machining'};
pie(piedata,labels)
```

```
clear piedata labels;
title('Total Cubesat Cost');

% %---------------power consumption UI-----------------------------
fig2=figure('Name','Power'); %give the fig a name and variable
set(fig2,'Position',[50 300 400 400]); %bottom,left,width,height
% batt+solar+prop+thruster+material+machining
piedata=[b(2,5)*10,b(2,6)*10,b(1,16)*10,b(1,17)*10];
labels = {'thruster','com','ADS','Payload'};
pie(piedata,labels)
clear piedata labels;
title('Power Consumption Breakdown');
```

```
% constraint.m constraints for the problem
function [C,Ceq]=constraint(X)

%--------Calculations--------------------------------
%--------Instructions: copy "objective" calculations
%---------into this section so that s,b,p can be
%---------calculated
global P

%---this section for gradient method--------
%---make sure to update "fixed" variables---
if length(X)==3
fixed=[P(4:6)]; %populate fixed variables from global P
vari=[X];
X=[fixed,vari];
end

%--------calculate y vector first time -----------------
% [battmass,propmass,proppwr]
y0=[.010 .400 50 100];   %set initial y values
y=y0;                 %declare y to be initial
[s]=structure(X,y); %run structure
y(4)=s(3,1);
[p]= prop(X,y);     %run prop
```

```
y(2)=p(3,4);         %update y, prop
y(3)=p(3,2);
[b] = battery(X,y);%run battery
yprev=y(1);
yprev2=y(2);
yprev3=y(3);
y(1)=b(3,4);            %update y, batt
ydiff=abs(y(1)-yprev);  %calculate ydiff
ydiff2=abs(y(2)-yprev2);
ydiff3=abs(y(3)-yprev3);
i=1;
diffplot(i)=ydiff;
diffplot2(i)=ydiff2;
diffplot3(i)=ydiff3;

%--------iterate the y vector for convergence---------
while (ydiff>0.001 || ydiff2>0.001 || ydiff3>0.001)
    i=i+1;
    [p]= prop(X,y); %run prop
    y(2)=p(3,4);         %update y
    y(3)=p(3,2);
    [b] = battery(X,y);%run battery
    yprev=y(1);             %save y as "previous"
    yprev2=y(2);
    yprev3=y(3);
    y(1)=b(3,4);            %update y
    ydiff=abs(y(1)-yprev);  %update ydiff
    ydiff2=abs(y(2)-yprev2);
    ydiff3=abs(y(3)-yprev3);
    diffplot(i)=ydiff;      %save each ydiff for plotting
    diffplot2(i)=ydiff2;
    diffplot3(i)=ydiff3;
end

% %----------Constraints--------------
% 1) power requirements met by solar + batteries
% 2) moment*modulus > baseline
% ...propellant in tanks is enough to satisfy delta-V
C1= b(3,1)-(b(2,2)+b(3,5));
C2= P(12)*10-s(3,3)*10;
C=[C1 C2];
```

```
Ceq=[];  %Ceq must be empty while using integer vars
end
```

```
% objective.m - runs the objective function

function [j,s,p,b,y,c] = objective(X)
global P
% lam=0.5;
%---this section for gradient method--------
%---make sure to update "fixed" variables---
if length(X)==3
fixed=[P(4:6)]; %populate fixed variables from global P
vari=[X];
X=[fixed,vari];
end

%--------calculate y vector first time -----------------
% [(1)battmass(2)prop mass(3)pwr draw,(4)struct mass]
y0=[.010 .400 50 100];   %set initial y values
y=y0;                 %declare y to be initial
[s]=structure(X,y); %run structure
y(4)=s(3,1);        %(structure mass)
[p]= prop(X,y);     %run prop
y(2)=p(3,4);         %update y, prop (thruster system mass)
y(3)=p(3,2);          %(total power draw)
[b] = battery(X,y);  %run battery
yprev=y(1);
yprev2=y(2);
yprev3=y(3);
y(1)=b(3,4);            %update y, batt
ydiff=abs(y(1)-yprev);  %calculate ydiff
ydiff2=abs(y(2)-yprev2);
ydiff3=abs(y(3)-yprev3);
i=1;
diffplot(i)=ydiff;
diffplot2(i)=ydiff2;
diffplot3(i)=ydiff3;

%--------iterate the y vector for convergence---------
```

```
while (ydiff>0.001 || ydiff2>0.001 || ydiff3>0.001)
    i=i+1;
    [p]= prop(X,y); %run prop
    y(2)=p(3,4);         %update y
    y(3)=p(3,2);
    [b] = battery(X,y);%run battery
    yprev=y(1);             %save y as "previous"
    yprev2=y(2);
    yprev3=y(3);
    y(1)=b(3,4);            %update y
    ydiff=abs(y(1)-yprev);  %update ydiff
    ydiff2=abs(y(2)-yprev2);
    ydiff3=abs(y(3)-yprev3);
    diffplot(i)=ydiff;      %save each ydiff for plotting
    diffplot2(i)=ydiff2;
    diffplot3(i)=ydiff3;
end

%------run the cost module-----------------------
[c]=cost(X,s,p,b);
%------plot the function updates for checking------
% figure
% plot(diffplot,'o');
% hold on
% plot(diffplot2,'+');
% plot(diffplot3,'*');

%---------calculate the objective value---------
mass=c(3,2)*1000; %now in grams
j=mass;
end
```

```
% structure.m calculates cubesat structure module
%-----------------------------------------------

%------module info-----------------------------
% base design has:
% t=2mm
% a=6.35mm
```

```
% starting from measured in solidworks
% base mass=149 (g)
% volume=42679 (mm^3)
% calculated mass= base structure mass*(A/baseA)
% constraint:
% modulus*cy >= base_modulus* base_cy

%--------begin function ------------------------
function [s] = structure(X,y)

%---------------assign variables----------------
s101 = X(3);          %material
s102 = 2.7;           %density, g/cc
s103 = 10*10*100/1000; %volume, l*w*h/100 = cc
s104=.002; % rail thickness, t (m)
s105=X(6); % rail width, a (m)
s201=0; % material density (kg/m^3)
s202=0; % machinability (%)
s203=0; % material cost ($/kg)
s204=0; % material modulus (GPa)
s205=0; % rail x-section area (m^2)
s301=0; % structure mass (m^2)
s302=0; % area moment of inertia (m^4)
s303=0; % moment*modulus (m^4*GPa)

%------------assign material data------------
%[density,machinability,cost/kg,modulus]
material=...
    [2700,360,18.08,68.9;...  %Al 6061 T6
     4500,22,141.11,113.8;...     %Titanium
     7850,72,5.47,200];  %A36 steel
for i=1:10;
    if s101==i;
s201   =  material(i,1);
s202   =  material(i,2);
s203   =  material(i,3);
s204   =  material(i,4);
    end
end
%----------calculations----------------------
%area=t*(2a-t) (m^2)
```

```
s205=s104*(2*s105-s104);
%area moment=(a^2+at-t^2)/(2*(2a-t)) (m^4)
s302=(s105^2+s104*s105-s104^2)/(2*(2*s105-s104));
%moment*modulus (m^4*GPa)
s303=s302*s204;
%mass=baseweight*area/basearea*density/basedensity (kg)
s301=0.149*(s205/2.14e-5)*(s201/2700);
%---------vectorize results-------------------
s=[s101 s102 s103 s104 s105;...
   s201 s202 s203 s204 s205;...
   s301 s302 s303 0.00 0.00];
end
```

```
% prop.m Cubesat Propulsion System (rev2)
%--------------------------------------------

function [p] = prop(X,y)
global P
p101=[5]; %initial disturbance (m/s)
p102=P(9); %disturbance per day (m/s/day)
p103=P(1); %mission duration (days)
p104=[4]; %thruster qty (#)
p105=[]; %cubesat mass (kg)
p106=X(1); %propellant type
p107=X(2); %thruster type
p108=P(10); %prop tank volume (ml)
p109=y(4); %structure mass (kg)
p110=[33e-3]; %propellant tank mass (kg)
p201=[]; % delta-V required (m/s)
%p202=[]; % impulse density of the fuel (N*s/kg)
p203=[]; % density of fuel (g/ml)
p204=[]; % specific impulse (s)
p205=[];% thruster mass (kg)
p206=[]; % thruster power (W)
p301=[]; % impulse required(N*s)
p302=[]; % total power draw (W)
p304=[]; % thruster system mass (kg)
p305=[]; % mass of propellant required (kg)
p306=[]; % volume of propellant required (ml)
p307=[]; % number of prop tanks (#)
```

```
% propellant
%[density(g/ml), ISP(s)]
propellant=...
 [0.02,296;... % Hydrogen
 0.04,179;... % Helium
 0.19,82;...% Neon
 0.28,80;...% Nitrogen
 0.44,57;...% Argon
 1.08,39;...% Krypton
 2.74,31;...% Xenon
 0.96,55;...% Freon 14
 0.19,114;...% Methane
 0.88,105];  %ammonia
%thrusters
%[mass(kg) power(W)]
thruster=...
[  .009,  1.000;... %Moog    (10W actually)
   .050,  0.090;... %Marotta (1W actually)
   .017,  0.890;... %Busek PFCV
   .035,  0.040];   %Lee LHDBO542115H
%assign propellant and thruster data
for i=1:10;
    if p106==i;
p203   =  propellant(i,1); %(density)
p204   =  propellant(i,2); %(ISP)
    end
    if p107==i;
p205  =  thruster(i,1); %(mass)
p206  =  thruster(i,2); %(power)
    end
end
%----------transient variables-------------
%total mass = accy mass + struct mass + batt mass + prop mass
p105=P(2)+y(4)+y(2)+y(1);
%----------calculations--------------------
%deltaV=initial+days*rate (m/s)
p201=p101+p102*p103;
%impulse req=deltaV*mass (N*s)
p301=p201*p105;
%propellant mass= I_required/(I_sp*gravity) (kg)
p305=p301/(p204*9.81);
```

```
%propvolume=mass/density (ml)
p306=p305*1000/p203;
%tanks required= volume/tankvolume (round up integer)
p307=ceil(p306/p108);
%mass=thruster qty*mass + proptank*mass
%     + fuelmass + structuremass(kg)
p304=p104*p205+p307*p110+p305+p109;
%power=thruster qty * thruster power (W)
p302=p104*p206;
%---------vectorize results----------------------
p=[p101, p102, p103, p104, p105, p106, p107, p108;...
   p201, 0.00, p203, p204, p205, p206, 0.00, 0.00;...
   p301, p302, 0.00, p304, p305, p306, p307, 0.00];
end
```

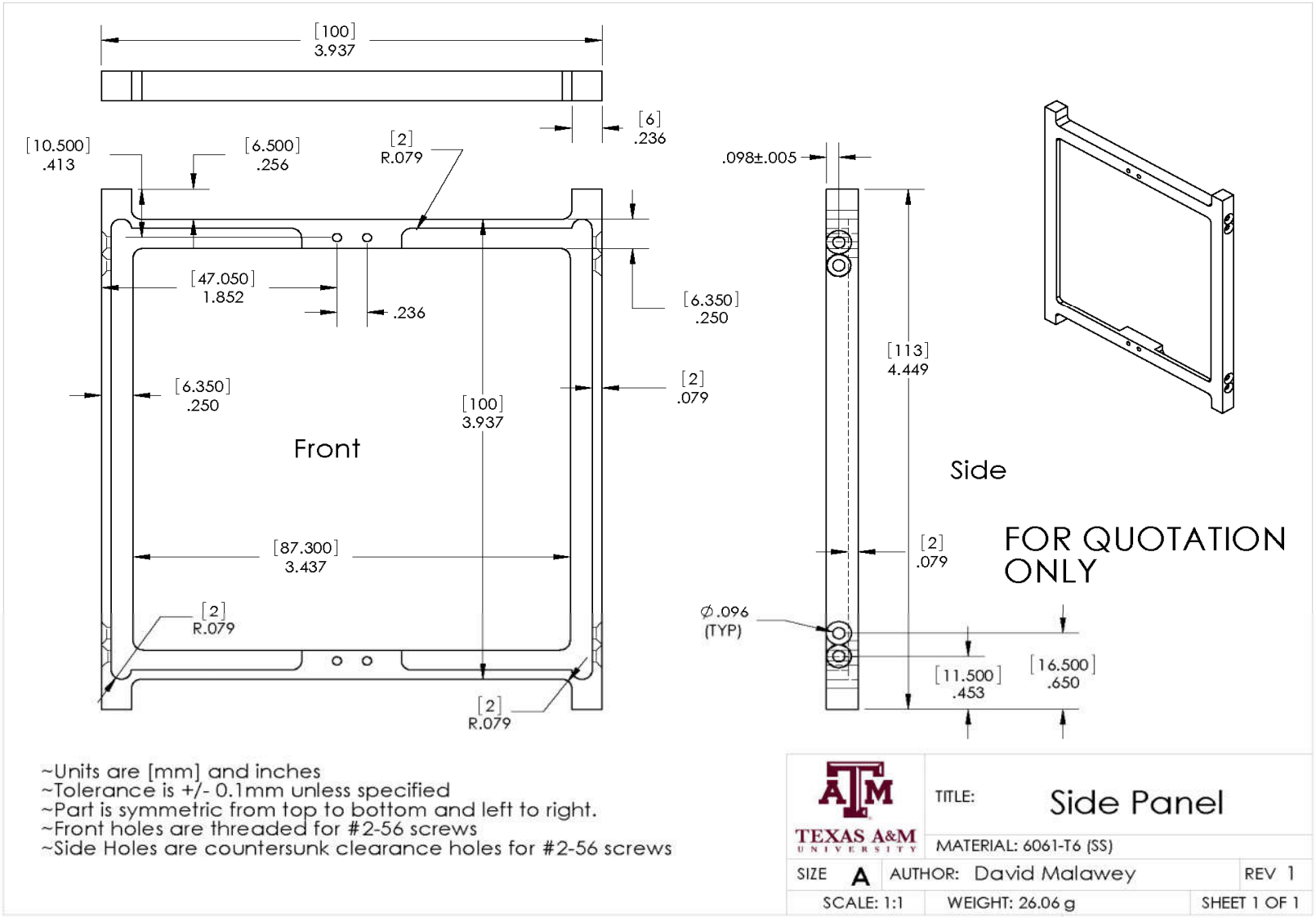
```
% battery.m Cubesat Power System
%-----------------------------------------------

function [b] = battery(X,y)

%---------LEVEL 1 VARIABLES--------------
global P
b101 = P(7); %battery capacity (mAh)
b102 = P(11); %factor of safety
b103 = P(1); %mission duration (days)
b104 = 300; % com power consumption (mA)
b105 = .10; % com duty ratio (ratio)
b106 = y(3); % prop power consumption (W)
b107 = .025;% prop duty ratio (ratio)
b108 = 44.5; %weight per battery (g)
b109 = P(8); % power of 1 solar panel (W)
b110 = 45;  % mass of 1 solar panel (g)
b111 = X(4); % number of solar panels (#)
b112 = 3.6; %battery voltage, nominal (v)
b113 = 3.3; %main circuit voltage (v)
b114 = X(5); % number of batteries (#)
b115= P(3); %solar panel sunlight(ratio)
b116= 0.04; %ADS average power (W)
b117= 0.02; %Payload avg power (W)
```

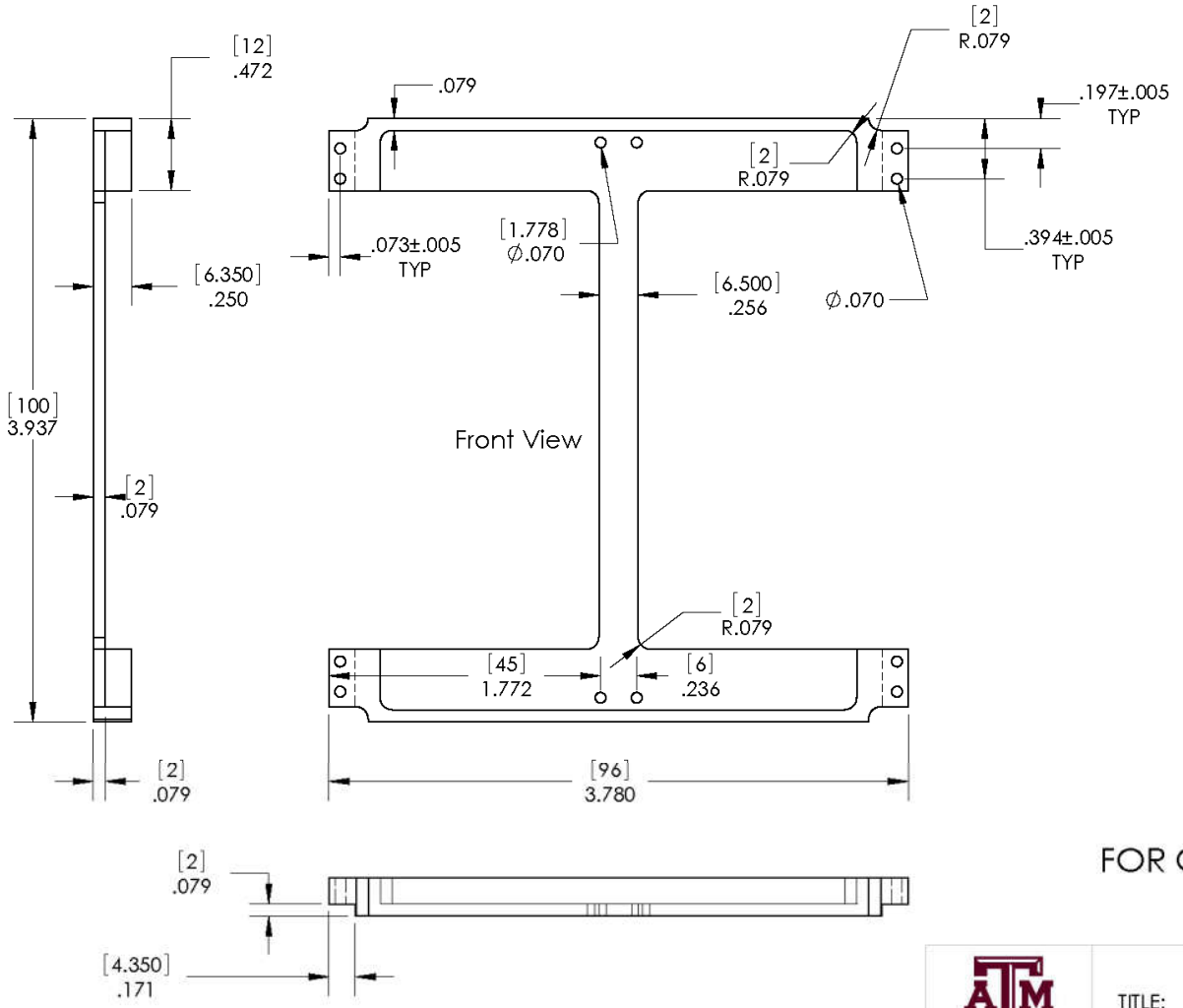
```
%-------------LEVEL 2 VARIABLES------------
b201 =0; %total mAh drawn (mAh)
b202 =0; %batteries energy (Wh)
b203=0; % energy used in misison, raw (Wh)
b204=0; % average power of mission, raw (W)
b205=0; % power used by thrusters
b206=0; % power used by com

%-------------LEVEL 3 VARIABLES------------
b301=0; % energy used in misison with FOS (Wh)
b302=0; % mass of batteries (g)
b303=0; % mass of solar panels (g)
b304=0; % mass, total power unit (g)
b305=0; % panels energy (Wh)

%-------------CALCULATIONS-----------------
%current drawn = miliamps * duration (mAh)
b201=(b104*b105)*b103*24;
%energy drawn = mAh/1000*v +prop pwr*duration(Wh)
%               +ADSpwr*h +Payloadpwr*h
b203=b201/1000*b113+(b106*b107*b103*24)...
        +b116*b103*24 +b117*b103*24;
%thruster power=W*ratio
b205=b106*b107;
%com power = mA/1000 * ratio * V
b206=b104/1000*b105*b113;
%power level = Wh/h
b204=b203/(b103*24);
%energy drawn with FOS = energy drawn *FOS (Wh)
b301=b203*b102;
%batt energy = qty*mAh*v/1000 (Wh)
b202=b114*(b101/1000)*b112;
%batt mass = mass * qty (g)
b302=b114*b108;
%panel mass * quantity (g)
b303=b110*b111;
%panel energy qty*watts*hours*sun_ratio (Wh)
b305=b111*b109*b103*24*b115;
% mass summation (kg)
b304=(b302+b303)/1000;
```

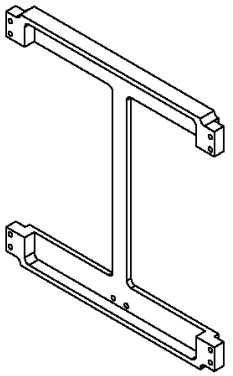
```
%-------------transient variables--------
b=[b101 b102 b103 b104 b105 b106 b107 b108 b109...
      b110 b111 b112 b113 b114 b115 b116 b117 0.00;...
      b201 b202 b203 b204 b205 b206 0.00 0.00 0.00...
      0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00;...
      b301 b302 b303 b304 b305 0.00 0.00 0.00 0.00...
      0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00];
end
```

```
% cost.m calculates the cost of the design

function [c] = cost(X,s,p,b)

%---------NAME VARIABLES--------------
global P
c101 = 10; % battery price ($)
c102 = 1500; % solar panel price ($)
c103 = []; % propellant price ($/kg)
c104 = []; % thruster price ($)
c105 = []; % material price ($/kg)
c106 = []; % material machinability (%)
c107 = 960; %cost to machine 6061, as base cost ($)
c201 =[]; %battery cost ($)
c202 =[]; %solar panel cost ($)
c203 =[]; %propellant cost ($)
c204 =[]; %thruster cost($)
c205 =[]; %material cost ($)
c206 =[]; %machining cost ($)
c301=[]; %total cost ($)
c302=[]; %total mass (kg)

%--------translate variables---------
% propellant cost ($/kg)
propellant=[120,52,330,4,5,330,1200,10,10,10];
    % Hydrogen, Helium, Neon, Nitrogen
    % Argon, Krypton, Xenon, Freon 14*
    % Methane*, ammonia* (* means guess)
%thrusters cost
thruster=[111,222,66,700];
% Moog 58X125A, marotta, Busek, Lee
```

```
% assign propellant and thruster data
% material cost ($/kg,machinability)
material=[18.08,360;...
          141.11,22;...
          5.47,72];
for i=1:10;
    if X(1)==i;
c103   =  propellant(i);
    end
    if X(2)==i;
c104  =  thruster(i);
    end
    if X(3)==i;
c105= material(i,1);
c106= material(1,2);
    end
end
%-------------CALCULATIONS-----------------
% batt cost = price * number of batteries
c201=c101*X(5);
% panel cost = price * number of solar panels
c202=c102*X(4);
% propellant cost = price * mass of propellant(kg)
c203=c103*p(3,5);
% thruster cost = price * number of thrusters
c204=c104*p(1,4);
% mass of material (kg)
c205=c105*1;
%machine cost = base cost % base_mch'ty/mch'ty
c206=c107*360/c106;
%total cost =batt+solar+prop+thruster+material+machining
c301=  c201+c202 +c203 +c204   +c205     +c206;
%total mass =
c302=b(3,4)+p(3,4)+s(3,1)+P(2);
%-------------vectorize results------------
c=[c101 c102 c103 c104 c105 c106 c107;...
   c201 c202 c203 c204 c205 c206 0.00;...
   c301 c302 0.00 0.00 0.00 0.00 0.00];
end
```


[100]
3.937
[6]
.236
[10.500]
.413
[6.500]
.256
[2]
R.079
[47.050]
1.852
.236
[6.350]
.250
[6.350]
.250
[2]
.079
[100]
3.937
Front
[87.300]
3.437
[2]
R.079
[2]
R.079
.098±.005
[113]
4.449
Side
[2]
.079
FOR QUOTATION ONLY
Ø.096
(TYP)
[11.500]
.453
[16.500]
.650
~Units are [mm] and inches
~Tolerance is +/- 0.1mm unless specified
~Part is symmetric from top to bottom and left to right.
~Front holes are threaded for #2-56 screws
~Side Holes are countersunk clearance holes for #2-56 screws
TEXAS A&M UNIVERSITY
TITLE: Side Panel
MATERIAL: 6061-T6 (SS)
SIZE A
AUTHOR: David Malawey
REV 1
SCALE: 1:1
WEIGHT: 26.06 g
SHEET 1 OF 1



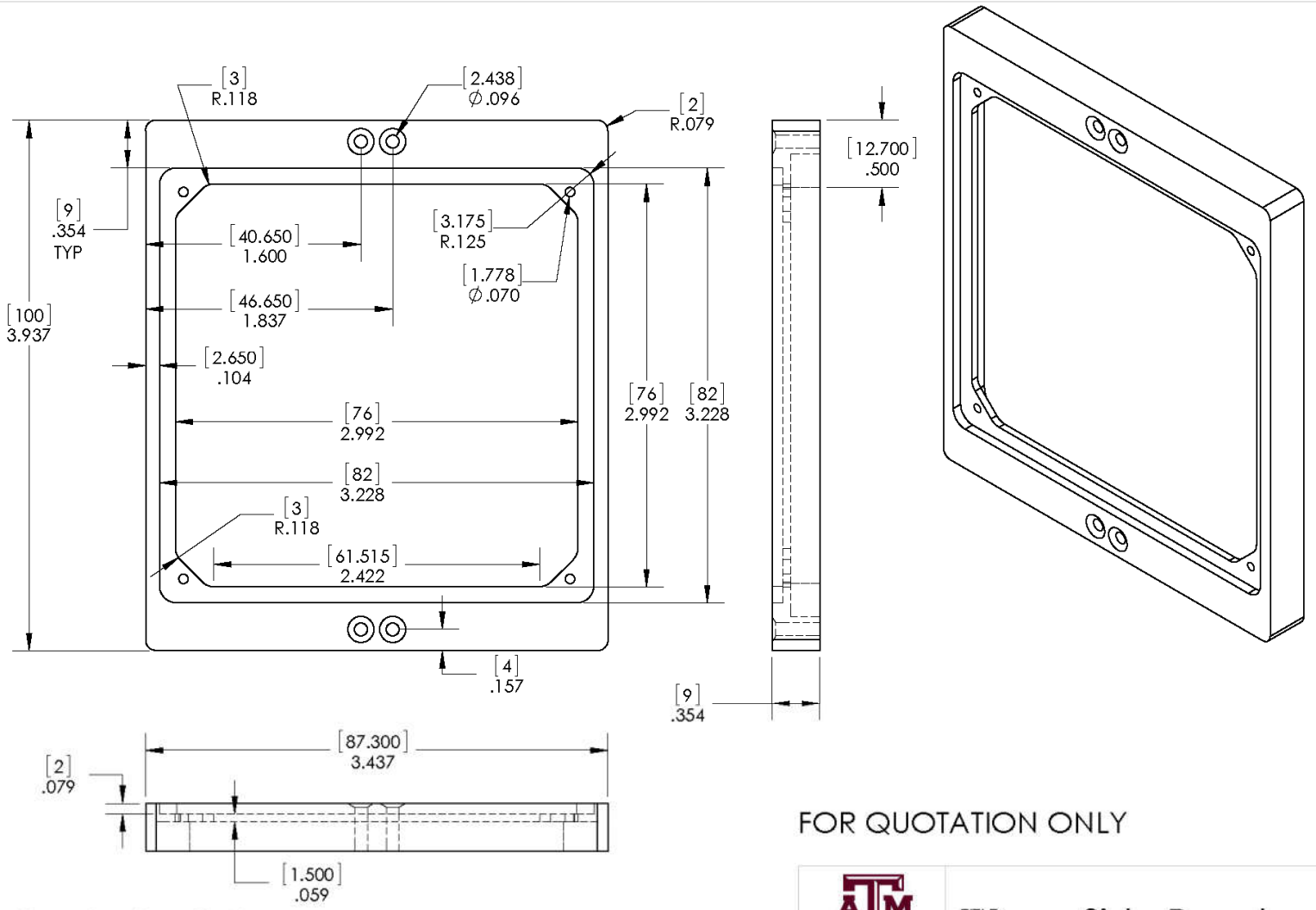


FOR QUOTATION ONLY

~Units are [mm] and inches
~Part is symmetric from left to right and top to bottom.
~Use Back face, Top, Bottom, and outside faces as reference (not front).
~Holes are size #50 with 2-56 threads

TEXAS A&M UNIVERSITY

TITLE: I Member		
MATERIAL: 6061-T6 (SS)		
SIZE A	AUTHOR: David Malawey	REV 1
SCALE: 1:1	WEIGHT: g21.52	SHEET 1 OF 1

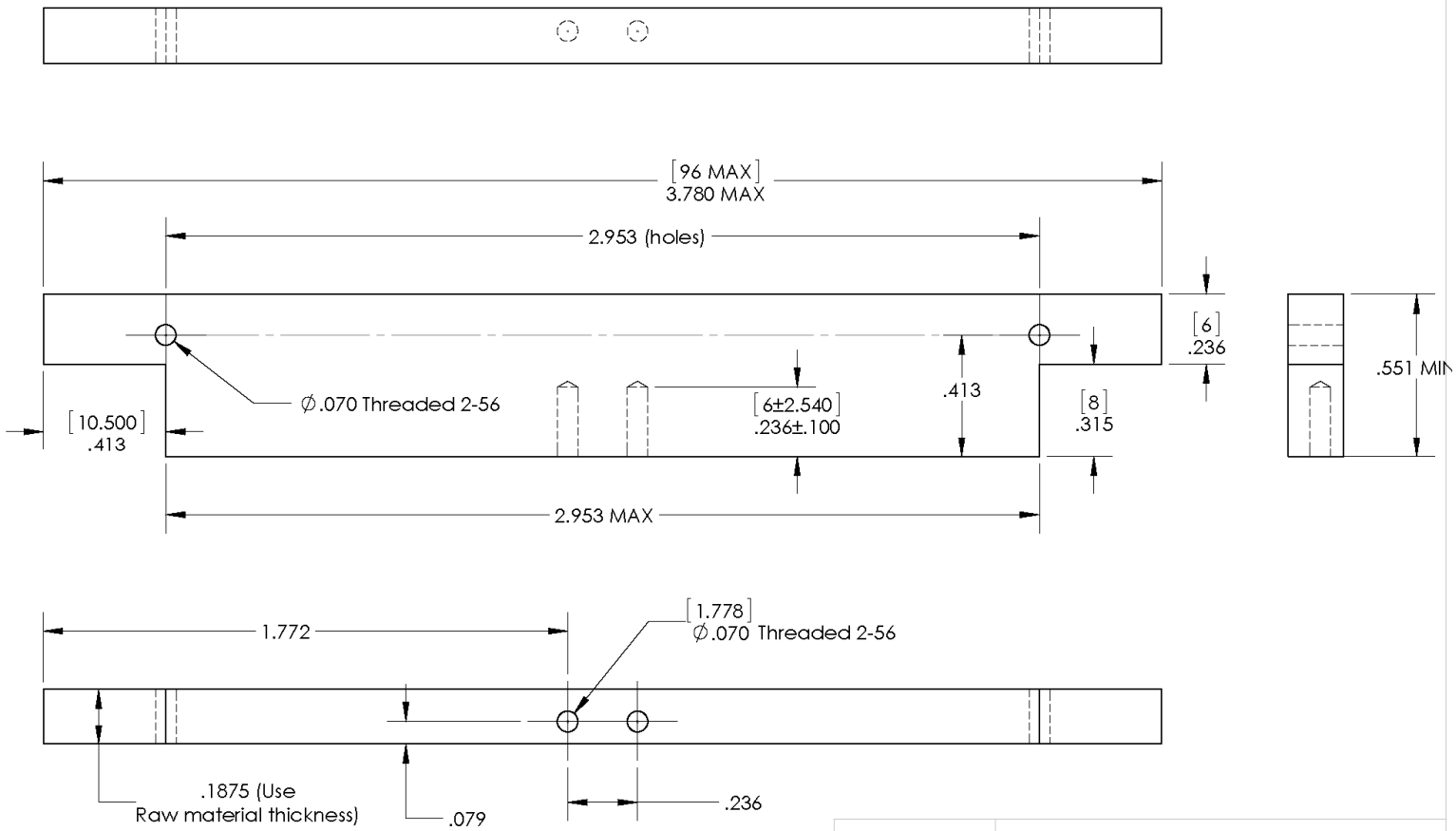


~Units are [mm] and inches
~Tolerance is +/- 0.1mm unless specified.
~Part is symmetric top to bottom and left to right.
~Large holes are #41 and countersunk.
~Small holes are #50, threaded holes for 2-56 screws.

FOR QUOTATION ONLY

TEXAS A&M UNIVERSITY	TITLE: Side Panel	
	MATERIAL: 6061-T6 (SS)	
SIZE A	AUTHOR: David Malawey	REV 1
SCALE: 1:1	WEIGHT: g44.68	SHEET 1 OF 1

Tolerance for hole location is 0.005 inch



	TITLE: NESI Mount Bracket	
	MATERIAL: 6061-T6 (SS)	
SIZE A	AUTHOR: David Malawey	REV 1
SCALE: 2:1	WEIGHT: g14.9752	SHEET 1 OF 1

APPENDIX C – FABRICATION MATERIALS

Supplier	PN	Description	Price (ea)	Purpose
Mcmaster carr	897 5k5 14	24 x 4 x 0.25" 6061 Aluminum rectangular bar	$20.66	cutting out cube panels
MSC	320 123 87	72 x 4 x 3/8" 6061 aluminum rectangular bar	$62.15	Cutting out fixed panels
MSC	744 597 28	Hertel drill bit #41 size 0.096 inch	$4.90	clearance holes for 2-56 screws
MSC	044 225 64	straight flute tap for 2-56 holes,	$6.81	tapping threads in holes
MSC	627 734 60	1/4" mill 1/4" shank, accupro, 2 flute	$14.42	cutting panels
MSC	88245980	1/8" mill 1/8" shank, hertel, 2 flute, carbide	$7.99	cutting panels
amazon.com	GP80*80-10A100	Sunnytech 0.5w mini solar panel	$6.99	cube solar panel
Mcmaster carr	927 03A 105	1/4" torx cntrsnk screws, stainless, torx, 2-56 (50pack)	$6.58	attach side panels to I-members
Home Depot	NA	2-56 Countersink drill bit	$11.88	countersinking holes
MSC	050 416 45	Tap Wrench	$16.37	holds the tap
MSC	40914699	1/2" phillips cntrsnk screw, 2-56, (50 pack)	$6.85	attaches the fixed panels
MSC	017 871 67	Engraving cutter	$16.81	cuts part ID on the panels
MSC	744 596 11	#50 size drill bit with 1/8" shank	$4.80	drilling tap holes for 2-56
MSC	320 122 96	Aluminum rectangular bar 1/4"x4"x72"	$42.83	stock for side panels & I-members
MSC	886 278 15	Shoulder screw, 1/4" #10-24 (course thread) qty5	$2.02	
Mcmaster carr	890 15k 49	Aluminum sheet 6"x48"x0.040"	$25.20	outer panels
amazon.com	NCR18650B	Panasonic lithium ion battery (18650 size), 4pk	$21.99	cube batteries
Future orders				
Mcmaster Carr	818 1K1 2	carbon fiber 1/32" sheet, 12x12"	36.58	fixed panels
Mcmaster Carr	181 3A2 43	2-part loctite epoxy for carbon fiber, loctite PN 9340	17.66	adhesive for carbon fiber
Mcmaster carr	736 1A1 1	0.005" thick 1/8" wide kapton polyimide tape (5yd)	5.62	adhesive for carbon fiber
Amazon.com	NA	Ernie Ball 2627 Beefy Nickel wound guitar strings	5.49	antenna wire